Title
Accounting for Calibration Uncertainty in Detectors for High-Energy Astrophysics

Permalink
https://escholarship.org/uc/item/0rg9z7d7

Author
Xu, Jin

Publication Date
2014

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

Accounting for Calibration Uncertainty in Detectors for High-Energy Astrophysics

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Statistics

by

Jin Xu

Dissertation Committee:
Associate Professor Yaming Yu, Co-Chair
Professor David van Dyk, Co-Chair
Associate Professor Zhaoxia Yu

2014

DEDICATION

This dissertation is lovingly dedicated to my parents, Caijuan Sun and Tingshang Xu.
Their constant support, deep encouragement, and eternal love have sustained me
throughout my life.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Prof. David van Dyk for the continuous support of my Ph.D study and research. Without his patience and guidance, I can not image how my Ph.D life would become. This 5-year experience of studying from David will be the most valuable treasure in my life.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Yaming Yu, and Prof. Zhaoxia Yu. Especial thanks to Yaming for the help in the thesis writing. Your encouragement and comments make the thesis possible.

My thanks also goes to my advisor at Adobe, William Yan. The offer for the summer internship and the following full-time position at Adobe open my new world after my Ph.D life. Your support and trust is my motivation to become a data scientist as my career.

I also thank my fellow friends, Shandong Zhao, Xu Tian, Yan He, Jie Shen, Bo Zhou, Shiwei Lan, David Stenning, Nicholas George, Guoqin Zheng, Hao Yang, Qingyun Li, and Lei Pei. You all make my life at UCI wonderful and colorful. Thank you.

Mostly, to my girl friend, Chenyu Shi, who loved and supported me during my most difficult time, and made me feel like anything was possible. I cannot image the life without you.

Last but not least, the funds are crucial for the whole work and reserach. This work was supported by NASA contract NAS8-03060 to the Chandra X-ray Center (VLK and AS), NSF grants DMS 09-07522, DMS-0907185, DMS-12-08791 and DMS-12-09232 (DvD, XLM, YY, AC, JX). In addition, DvD was supported in part by a British Royal Society Wolfson Research Merit Award, by a European Commission Marie-Curie Career Integration Grant, and by the STFC (UK).

CURRICULUM VITAE

Jin Xu

EDUCATION

Doctor of Philosophy in Statistics **2014**
University of California, Irvine *Irvine, California*

Bachelor of Science in Finance **2008**
Tsinghua University *Beijing, China*

RESEARCH EXPERIENCE

Graduate Research Assistant **2009–2014**
University of California, Irvine *Irvine, California*

- Illustrated the essential differences between upper bounds and upper limits terms and applied Bayesian methods in the calculation.
- Systematized MCMC methods in model fitting in order to incorporate calibration uncertainty and proved ignorance of calibration uncertainty can highly influence the results.
- Presented the significance of our findings to astronomers.

TEACHING EXPERIENCE

Teaching Assistant **2009–2014**
University of California, Irvine *Irvine, California*

JOURNAL PUBLICATIONS

Xu, J., van Dyk, D. A., Connors, A., Kashyap, V. L., Siemiginowska, A., and Yu, Y.. A Fully Bayesian Method to Simultaneously Fit Calibration Products and Model Parameters in X-ray Spectral Analyses. (submitted) **2014**
The Astrophysical Journal

Kashyap, V. L., van Dyk, D. A., Connors, A., Freeman, P. E., Siemiginowska, A., Xu, J., and Zezas, A.. On Computing Upper Limits to Source Intensities. **2010**
The Astrophysical Journal 719, 900-914.

Working Experience

Adobe Systems Incorporated **Jun-Sep 2013**
Data Scientist Summer Intern

- Investigated the people's behavior at Facebook Ads platform.
- Developed an algorithm to analyze the factors that affect the outcome (Click Through Rate) of Facebook Campaign.
- Developed a robust algorithm to make trend approximation for any arbitrary binned data.

Bank of China **Aug 2009**
Data Analyst Summer Intern

- Researched exportation and importation and presented results to the managers.
- Performed competitive analysis of local and international trade.

SOFTWARE

C, C++, Python, R, SAS, SPSS, STATA (exposure), BUGS, Winbugs, Matlab, Java, LATEX, SQL, Excel, Word, Outlook, Publisher

ABSTRACT OF THE DISSERTATION

Accounting for Calibration Uncertainty in Detectors for High-Energy Astrophysics

By

Jin Xu

Doctor of Philosophy in Statistics

University of California, Irvine, 2014

Associate Professor Yaming Yu, Co-Chair

Professor David van Dyk, Co-Chair

Systematic instrumental uncertainties in astronomical analyses have been generally ignored in data analysis due to the lack of robust principled methods, though the importance of incorporating instrumental calibration uncertainty is widely recognized by both users and instrument builders. Ignoring calibration uncertainty can cause bias in the estimation of source model parameters and can lead to underestimation of the variance of these estimates. Lee *et al.* (2011) introduced a so-called pragmatic Bayesian method to address this problem. The method is pragmatic in that it introduces an ad hoc technique that simplifies computation by assuming that the current data is not useful in narrowing the uncertainty for the calibration product, i.e., that the prior and posterior distributions for the calibration products are the same.

In the thesis, we focus on incorporating calibration uncertainty into a principled Bayesian X-ray spectral analysis, specifically we account for uncertainty in the so-called effective area curve and the photon redistribution matrix. X-ray spectral analysis models the distribution of the energies of X-ray photons emitted from an astronomical source. The effective area curve of an X-ray detector describes its sensitive as a function of the energy of incoming photons, and the photon redistribution matrix describes the probability distribution of the recorded (discrete) energy of a photon as a function of the true (discretized) energy. Starting with the effective area curve, we follow Lee *et al.* (2011) and use a principle component analysis (PCA) to efficiently represent the uncertainty. Here, however, we leverage this representation to enable a principled, fully Bayesian method to account for calibration uncertainty in high-energy spectral analysis. For the photon redistribution matrix, we first model each conditional distribution as a normal distribution and then apply PCA to the parameters describing the normal models. This results in an efficient low-dimensional summary of the uncertainty in the redistribution matrix. Our methods for both calibration products are compared with standard analysis techniques and the pragmatic Bayesian method of Lee *et al.* (2011). The advantage of the fully Bayesian method is that it allows the data to provide information not only for estimation of the source parameters but also for the calibration product; we demonstrate this for the effective area curve. In this way, our fully Bayesian approach can yield more accurate and efficient estimates of the source parameters, and valid estimates of their uncertainty. Moreover, the fully Bayesian approach is the only method that allows us to make a valid inference about the effective area curve itself, quantifying which possible curves are most consistent with the data.

Chapter 1

Background

In this chapter, we will introduce the background about calibration and systematic errors in the physical sciences, and the important role that calibration plays in drawing valid scientific conclusions. To account for calibration uncertainty, we will adopt Bayesian methods throughout the entire thesis. Basic inference and computation techniques of Bayesian approaches will be reviewed in this chapter. Since we will use Principle Component Analysis (PCA) to quantify calibration uncertainty, the concept of PCA will also be reviewed in this chapter.

1.1 Calibration and Systematic Errors

In physical sciences and other sciences, measurement error, the difference between a measured value of quantity and its true value, is very common and in fact unavoidable in practice. The statistical approach for measurement errors can vary enormously, depending on different scenarios and different factors. Measurement errors can be divided into two categories: random error and systematic error. Random errors are statistical fluctuations (in either direction) in the measured data due to the precision limitations of the measurement device. Random errors usually result from the experimenter's inability to take the same measurement in exactly the same way to get the exact same value. Systematic errors, by contrast, are reproducible inaccuracies that are consistent in the same direction. Systematic errors are often due to a problem which is inherent in the measurement process or the measuring instruments.

A systematic error is sometimes called statistical bias. It cannot be entirely removed due to the imperfection of the measuring instruments. Systematic errors can be either constant, or be dependent (e.g. as a proportion or a percentage) on the actual value of the measured quantity, or even be related to the value of a different quantity of the environment (e.g. the reading of a ruler can be affected by the temperature).

If a systematic error is constant, a common method to reduce it is through calibration of the measuring instrument. In practice, a systematic error can possibly be a random quantity. In this case, simple constant calibration for the measuring instrument can no longer be applied. How to appropriately account for this *calibration uncertainty* is the main goal in this thesis.

There is a large literature on measurement errors. In physical sciences, Moffat (1988), Dieck (1992) and Lira (2002) described the basic source of uncertainties in experiments and the methods to account for measurement errors. Weise and Woger (1993) pointed out a maximum-entropy based method to account for measurement errors. This method without parameterizing the measure error source can be applied in repeated experiments. In astrophysics, measurement errors and calibration methods are also widely discussed. Kelly (2007) described some aspects of measurement error in linear regression of astronomical data, and Kutner and Ulich (1981) recommended reliable relative and absolute intensity calibration schemes for millimeter-wavelength observations of spectral line. How to count for different sources of measure errors in data calibration is crucial in various sciences, and ignoring the measure errors may lead to substantial bias in research.

Let us use a simple example to illustrate the general ideas of systematic errors. Physical engineers measure temperatures using thermocouples. The variability in these measurements comes from two sources: (1) the devices are not perfectly calibrated, some read too high, some too low; and (2) the devices are not perfectly consistent either – repeated measurements with the same thermocouple might result in slightly different readings. In reality, the engineers do not use a new random thermocouple for each reading they make. Instead, one thermocouple is used to make multiple readings. This converts the first type of error from random error into systematic error. The second type of error remains random error.

If the chosen thermocouple always reads too high, the temperature measurement will be overestimated, and vice versa. Thus ignoring systematic error can lead to biased measurements.

Consider a simple statistical model for measuring temperature using one thermocouple.

$$Z_{obs} = Z_{true} + C + \epsilon, \tag{1.1}$$

where Z_{obs} is the observed measurement, Z_{true} is the true temperature value, C is the calibration uncertainty, and ϵ is the random error. Here, we simply use additivity to represent the relationship among the true temperature value, calibration uncertainty and the observed value. In reality, as will be discussed in the context of high-energy astrophysics, this relationship could be very complicated.

1.2 Bayesian Inference

In order to appropriately account for calibration uncertainty, we adopt a Bayesian approach. This approach gives us a sound mathematical framework for combining the data with external sources of information. In particular, we quantify our state of knowledge before having seen the data using a *prior distribution* and that after having seen the data using a *posterior distribution*. Bayes' Theorem allows us to transform the prior distribution into the posterior distribution using the observed data.

In particular, suppose Y is the observed data, ψ represents the unknown parameters, and $\mathcal{I}$ represents the information we have before seeing the data. In this setting, Bayes' Theorem states that the posterior distribution of ψ given Y and $\mathcal{I}$ is:

$$p(\psi|Y, \mathcal{I}) = \frac{L(Y|\psi, \mathcal{I})\pi(\psi|\mathcal{I})}{p(Y|\mathcal{I})}, \tag{1.2}$$

where $\pi(\psi|\mathcal{I})$ is the prior distribution of ψ, $L(Y|\psi, \mathcal{I})$ is the likelihood of Y given ψ, and $p(Y|\mathcal{I})$ represents the marginal distribution of Y, that is $p(Y|\mathcal{I}) = \int p(Y|\psi, \mathcal{I})\pi(\psi|\mathcal{I})d\psi$ is a normalizing constant that ensures $p(\psi|Y, \mathcal{I})$ integrates to one.

Let us consider the temperature measurement model of Section 1.1. In Equation (1.1), we have $Y = Z_{obs}$ and we treat Z_{true}, and C as unknown quantities, that is $\psi = (Z_{true}, C)$. We will need external information from physics scientists to set up prior $\pi_1(Z_{true}|\mathcal{I})$ for the true temperature value Z_{true}, and external information from calibration scientists to set up the prior $\pi_2(C|\mathcal{I})$ and $\pi_3(\epsilon|\mathcal{I})$ for C and the density of the random error ϵ. Here we assume, that the priors for Z_{true} and $C$are independent of each other. Combining Equation (1.1), we have the conditional probability distribution of Z_{obs}, $p(Z_{obs}|Z_{true}, C, \mathcal{I}) = \pi_3(Z_{obs} - Z_{true} - C|\mathcal{I})$

Using Bayes Theorem, we have the posterior distribution of Z_{true} and C:

$$p(Z_{true}, C|Z_{obs}) = \frac{L(Z_{obs}|Z_{true}, C)\pi_1(Z_{true})\pi_2(C)}{p(Z_{obs})}, \tag{1.3}$$

where $L(Z_{obs}|Z_{true}, C)$ is the likelihood of Z_{obs} given Z_{true}, and C, that is $L(Z_{obs}|Z_{true}, C) = \pi_3(Z_{obs} - Z_{true} - C)$ and $p(Z_{obs}) = \int L(Z_{obs}|Z_{true}, C)\pi_1(Z_{true})\pi_2(C)dZ_{true}dC$ is the marginal distribution of Z_{obs}. In the remainder of this chapter, we omit $\mathcal{I}$ in our notation.

As a simple illustration, we set up the distribution of Z_{true}, C and ϵ as normal distributions, centered at zero,

$$\begin{aligned}
&Z_{true} \sim \mathcal{N}(0, \sigma_Z^2) \\
&C \sim \mathcal{N}(0, \sigma_C^2) \\
&\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)
\end{aligned} \tag{1.4}$$

Here, we introduce three methods to make inference about Z_{true}, which we investigate in the later chapters.

The first method is called the Standard method. This method assumes that the true calibration uncertainty is known, equal to the default calibration uncertainty, that is $C = C_{def}$. Usually, C_{def} is simply set to zero, which means we believe that there is no bias. We have the posterior distribution of Z_{true} given C and Z_{obs}:

$$p_{\text{std}}(Z_{true}|C = 0, Z_{obs}) = \mathcal{N}\left(\frac{\sigma_Z^2}{\sigma_Z^2 + \sigma_\epsilon^2} Z_{obs}, \frac{\sigma_Z^2 \sigma_\epsilon^2}{\sigma_Z^2 + \sigma_\epsilon^2}\right) \tag{1.5}$$

The Standard method is simple as it does not incorporate any calibration uncertainty. In practice, this assumption is often suspect and will lead to biased results.

The second method is called the Pragmatic Bayesian Method. It was first introduced by Lee *et al.* (2011) in astrophysics. Mathematically, we have

$$p(Z_{true}, C|Z_{obs}) = p(Z_{true}|C, Z_{obs}) p(C|Z_{obs}),$$

The pragmatic Bayesian method makes the simplified assumption that $p(C|Z_{obs}) = \pi_2(C)$. This assumption means that the observed data carry no information about the calibration uncertainty. Under this assumption, we have the marginal distribution of Z_{true}:

$$p_{\mathrm{pB}}(Z_{true}|Z_{obs}) = \int p(Z_{true}|C, Z_{obs})\pi_2(C)dC,$$

After simple calculations, we have:

$$p_{\mathrm{pB}}(Z_{true}|Z_{obs}) = \mathcal{N}\left(\frac{\sigma_Z^2}{\sigma_Z^2+\sigma_\epsilon^2}Z_{obs}, \frac{\sigma_Z^2\sigma_\epsilon^2}{\sigma_Z^2+\sigma_\epsilon^2} + \left(\frac{\sigma_Z^2}{\sigma_Z^2+\sigma_\epsilon^2}\right)^2\sigma_C^2\right) \tag{1.6}$$

The pragmatic Bayesian method does not allow the data to choose the most suitable calibration uncertainty. As a result, $p_{\mathrm{pB}}(Z_{true}|Z_{obs})$ has the same mean location as standard method, but the variance is increased considerably.

The third method that we propose in this thesis is the Fully Bayesian method. It is a principled Bayesian method that treats Z_{true} and C as unknown quantities, both of which can be learned from the data. Based on Equations (1.1), (1.3), and (1.4), we have the joint posterior distribution of Z_{true} and C given Z_{obs}, $p(Z_{true}, C|Z_{obs})$, in closed from:

$$\left(\begin{array}{c} Z_{true} \\ C \end{array}\right)\Bigg| Z_{obs} \sim \mathcal{N}\left(\left(\begin{array}{c} \frac{\sigma_Z^2}{\sigma_Z^2+\sigma_C^2+\sigma_\epsilon^2}Z_{obs} \\ \frac{\sigma_C^2}{\sigma_Z^2+\sigma_C^2+\sigma_\epsilon^2}Z_{obs} \end{array}\right), \left(\begin{array}{cc} \frac{(\sigma_C^2+\sigma_\epsilon^2)\sigma_Z^2}{\sigma_Z^2+\sigma_C^2+\sigma_\epsilon^2} & -\frac{\sigma_C^2\sigma_Z^2}{\sigma_Z^2+\sigma_C^2+\sigma_\epsilon^2} \\ -\frac{\sigma_C^2\sigma_Z^2}{\sigma_Z^2+\sigma_C^2+\sigma_\epsilon^2} & \frac{(\sigma_Z^2+\sigma_\epsilon^2)\sigma_C^2}{\sigma_Z^2+\sigma_C^2+\sigma_\epsilon^2} \end{array}\right)\right),$$

which yields the marginal distribution of Z_{true}:

$$p_{\mathrm{fB}}(Z_{true}|Z_{obs}) = \mathcal{N}\left(\frac{\sigma_Z^2}{\sigma_Z^2+\sigma_C^2+\sigma_\epsilon^2}Z_{obs}, \frac{(\sigma_C^2+\sigma_\epsilon^2)\sigma_Z^2}{\sigma_Z^2+\sigma_C^2+\sigma_\epsilon^2}\right). \tag{1.7}$$

This simple example has an analytical solution and it illustrates how the fully Bayesian analysis can account for calibration uncertainty. When we treat calibration uncertainty as another unknown quantity, the observed data themselves can help us make inference about the calibration uncertainty and the unknown quantity of interest (e.g. temperature in this example) simultaneously. This is a great advantage of Bayesian analysis. In the following chapters, we will explore more in a more complex, realistic situation of the astrophysics.

1.3 Comparison of the Three Methods

Random errors can often be reduced by taking multiple measurements. For the temperature measuring example, we can measure the temperature multiple times and use the mean of temperature measures as Z_{obs} to reduce the random error ϵ. Thus, in order to make an intuitive comparison of these three methods, we assume that $\sigma_Z^2, \sigma_C^2 \gg \sigma_\epsilon^2$, that is, the random error variance σ_ϵ^2 is much smaller than prior variance σ_Z^2 and calibration uncertainty variance σ_C^2. We make the following comparison.

Table 1.1 summarizes the marginal distribution of Z_{true} of the three methods, with the assumption $\sigma_Z^2, \sigma_C^2 \gg \sigma_\epsilon^2$. We can see that the marginal distribution of Z_{true} from the pragmatic Bayesian method has the same mean as that from the standard method, while the mean of Z_{true} from the fully Bayesian is shifted. The pragmatic Bayesian has the biggest variance for Z_{true} , the standard method has the smallest variance, and the fully Bayesian is in-between.

Suppose that the true value of calibration uncertainty is C_{true}. We can evaluate the estimators of these three methods by *Bias*, *Variance*, and the mean squared error (*MSE*).

$$Bias(\hat{Z}_{true}) = E(\hat{Z}_{true} - Z_{true})$$

$$MSE(\hat{Z}_{true}) = E(\hat{Z}_{true} - Z_{true})^2 = Var(\hat{Z}_{true}) + (Bias(\hat{Z}_{true}, Z_{true}))^2$$

Table 1.2 illustrates the evaluation of these three methods for this example. The fully

Methods	Mean	Variance
Standard	Z_{obs}	σ_ϵ^2
Pragmatic Bayes	Z_{obs}	σ_C^2
Fully Bayes	$\frac{\sigma_Z^2}{\sigma_Z^2+\sigma_C^2} Z_{obs}$	$\frac{\sigma_C^2\sigma_Z^2}{\sigma_Z^2+\sigma_C^2}$

Table 1.1: Comparison of the marginal distribution of Z_{true}. For this simple temperature measuring example, the marginal distribution of Z_{true} from the pragmatic Bayesian method has the same mean as that from the standard method, while the mean of Z_{true} from the fully Bayesian has shifted. The pragmatic Bayesian has the biggest variance of Z_{true} , the standard method has the smallest variance, and the fully Bayesian is in-between.

Bayesian method has the smallest bias, compared to the standard method and the pragmatic Bayesian method. Since the standard method operates under the assumption of an incorrectly specified calibration uncertainty, it has very small $Var(\hat{Z}_{true})$and is valid only when C_{true} equals to zero or close to zero. In reality, if we have little prior information about the true calibration uncertainty, the standard method could lead to a biased and dangerous result. The pragmatic Bayesian method incorporates calibration uncertainty, leading to a larger $Var(\hat{Z}_{true})$, which correctly accounts for the variability of $\hat{Z}_{true}$, and makes the inference more valid than the standard method. However, if we compare the pragmatic Bayesian and the fully Bayesian, the fully Bayesian not only reduces the bias, but also reduces the MSE. Therefore, we recommend to use the fully Bayesian, if the true calibration uncertainty is unknown.

Methods	$Bias(\hat{Z}_{true})$	$Var(\hat{Z}_{true})$	$MSE(\hat{Z}_{true})$
Standard	C_{true}	σ_ϵ^2	$C_{true}^2 + \sigma_\epsilon^2$
Pragmatic Bayes	C_{true}	σ_C^2	$C_{true}^2 + \sigma_C^2$
Fully Bayes	$\frac{\sigma_Z^2}{\sigma_Z^2+\sigma_C^2}C_{true}$	$\frac{\sigma_C^2\sigma_Z^2}{\sigma_Z^2+\sigma_C^2}$	$(\frac{\sigma_Z^2}{\sigma_Z^2+\sigma_C^2}C_{true})^2 + \frac{\sigma_C^2\sigma_Z^2}{\sigma_Z^2+\sigma_C^2}$

Table 1.2: Evaluation of the three methods. In general, the standard method only works well when C_{true} is equal to zero or close to zero. Otherwise it results in biases. The pragmatic Bayesian is most conservative, and has the largest $Var(\hat{Z}_{true})$. Compared to the pragmatic Bayesian, the fully Bayesian not only reduces the bias, but also has a smaller MSE.

1.4 Fitting Bayesian Methods Using MCMC

The fundamental Equation (1.2) is often written as,

$$p(\psi|Y) \propto L(Y|\psi)\pi(\psi), \tag{1.8}$$

where we have omitted the denominator of Equation (1.2), a normalizing constant which might be difficult (and unnecessary) to compute. The proportionality sign, "$\propto$", signifies that the density function is not normalized.

To compute fitted parameters and their error bars in a Bayesian analysis, we use quantities such as the mean, variance and percentiles of the posterior distribution. Although in some simple cases (such as the temperature measuring example of Section 1.2), the posterior distribution is a well-known distribution and these quantities can be calculated analytically, numerical methods are typically used. One very popular numerical method is Monte Carlo which involves simulating draws from the posterior distribution and using the means, variances and percentiles of the Monte Carlo samples to compute fitted parameters and their error bars. Park *et al.* (2008) reviews how this can be done using examples from high-energy astrophysics.

One of the common Monte Carlo methods is Markov chain Monte Carlo (MCMC), where a Markov chain is constructed as an ordered sequence $\{\psi^{(t)}, t = 1, 2, ...\}$, each value of which only depends on the value of the previous variable in the chain. Specifically, $\psi^{(t+1)}$ is obtained by simulating from *transition kernel*, which is a conditional density that preserves the target posterior distribution. That is, if $\psi^{(t)}$ has the posterior distribution already, then so does $\psi^{(t+1)}$.

$$\psi^{(t+1)} \sim \mathcal{K}(\psi|\psi^{(t)}), \tag{1.9}$$

Here, the kernel $\mathcal{K}$ should be designed to be simple to sample from and ensure that the chain converges to the target distribution. Iterating Equation (1.9) results in a chain of values that is called a Markov chain, which delivers a correlated sample from the posterior distribution. Ergodic theory says that the chain will converge, that means $\psi^{(t)}$ has approximately posterior distribution, if t is large. In practice we must start the chain at perhaps some arbitrary value and let it run until it converges to the posterior distribution. This typically requires convergence diagnostics, such as running multiple chains until they appear to be sampling from the same distribution. This method is described in detail in Gelman and Rubin (1992), see also Brooks and Gelman (1998) and Brooks *et al.* (2011).

The Metropolis-Hastings algorithm (Metropolis *et al.* (2004) and Hastings (1970)) and the Gibbs sampler (Gelfand and Smith (1990) and Geman and Geman (1984)) are the two popular MCMC algorithms. They correspond to specific kernels in Equation (1.9) to ensure that the resulting Markov chain converges properly to the target posterior distribution. Here, we provide some background about the algorithms that we will apply in the later chapters.

The Metropolis-Hastings algorithm draws samples for any probability distribution $p(\psi|Y)$, provided that one can compute the unnormalized target density. That is, we only need to calculate $L(Y|\psi)\pi(\psi)$ to apply Metropolis-Hastings algorithm. This makes the Metropolis-

Hastings algorithm particularly useful, because calculating the normalized constant is often extremely difficult in practice. The Metropolis-Hastings algorithm uses a proposal distribution or jumping distribution $Q(\psi|\psi^{(t)})$ to suggest a candidate for the next sampled value given the previously sampled value. This proposed value is then accepted or rejected according to a specific ratio, which ensures that the chain converges to the correct distribution.

Metropolis-Hastings Sampler:

For $t = 0, 1, 2, \dots, T$,

Step 1: Simulate $\psi^{\text{prop}} \sim Q(\psi \mid \psi^{(t)})$ and compute

$$\alpha = \frac{p(\psi^{\text{prop}} \mid Y)Q(\psi^{(t)} \mid \psi^{\text{prop}})}{p(\psi^{(t)} \mid Y)Q(\psi^{\text{prop}} \mid \psi^{(t)})}.$$

Step 2: Let u be a uniformly distributed random number between zero and one and set

$$\psi^{(t+1)} = \begin{cases} \psi^{\text{prop}} & \text{if } u < \alpha \\ \psi^{(t)} & \text{otherwise} \end{cases}.$$

This algorithm proceeds by randomly attempting to move about the sample space, sometimes accepting the moves and sometimes remaining at the previous iteration.

The Metropolis Sampler is a special case of the Metropolis-Hastings Sampler, when we have a symmetric proposal in the sense that, $Q(\psi^{\text{prop}} \mid \psi^{(t)}) = Q(\psi^{(t)} \mid \psi^{\text{prop}})$. The acceptance ratio is simplified into $\alpha = p(\psi^{\text{prop}} \mid Y)/p(\psi^{(t)} \mid Y)$. If the proposed move is to a point that is more probable than the existing point (i.e. a point in a higher-density region of $p(\psi|Y)$), we will always accept the move. However, if the proposed move is to a less probable point, we will sometimes reject the move, and the bigger the relative drop is in probability, the more likely we are to reject the new point. Thus, we will tend to stay in (and return large numbers of samples from) high-density regions of $p(\psi|Y)$, while only occasionally visit low-density regions. Because of the symmetric property of proposal distribution, the Metropolis Sampler is also called as a Random Walk Sampler.

The Independence Sampler is another special case of the Metropolis-Hastings Sampler, when we set proposal distribution to be independent from the current state, that is $Q(\psi^{\text{prop}} \mid \psi^{(t)}) = Q(\psi^{\text{prop}})$. Getting efficient samples, the proposal distribution needs to be as close to the target distribution as possible. In the later chapters, we will use both the Metropolis Sampler and the Independence Sampler.

The Gibbs sampler is widely used when direct sampling from a multivariate distribution is difficult. The Gibbs Sampler is an iterative algorithm, where at each iteration, we simulate one component of a parameter vector at a time, according to the conditional distribution of that component given all other components. It is especially useful when each full conditional distribution is available. For instance, suppose ψ contains two components, $\psi = (\upsilon, \omega)$. We can apply a two-step Gibbs sampler.

Two Step Gibbs Sampler

For $t = 0, 1, 2, \ldots, T$,

Step 1: Simulate $\upsilon^{(t+1)} \sim p(\upsilon|\omega^{(t)}, Y)$

Step 2: Simulate $\omega^{(t+1)} \sim p(\omega|\upsilon^{(t+1)}, Y)$

Gibbs sampler ensures that the samples $\psi^{(t)} = (\upsilon^{(t)}, \omega^{(t)})$ can approximate the the joint distribution $p(\upsilon, \omega|Y)$. Furthermore, the marginal distribution of any variable can be approximated by simply examining the samples for this variable, ignoring the other variable.

The Partially Collapsed Gibbs (PCG) sampler is a modification on the Gibbs Sampler, which involves reducing the conditioning in one or more steps within an iteration (van Dyk and

Park, 2008). For example, if we can easily sample from the marginal distribution $p(\upsilon|Y)$, we can obtain a PCG sampler from the Gibbs sampler. PCG samplers typically converge faster than their parent Gibbs samplers.

PCG Sampler

For $t = 0, 1, 2, \ldots, T$,

Step 1: Simulate $\upsilon^{(t+1)} \sim p(\upsilon|, Y)$

Step 2: Simulate $\omega^{(t+1)} \sim p(\omega|\upsilon^{(t+1)}, Y)$

Since we remove the conditioning in the first step, we get independent draws $(\upsilon^{(t)}, \omega^{(t)})$, which makes PCG converge to the stationary distribution $p(\upsilon, \omega|Y)$ immediately. Notice this example is only one case of PCG, and this thesis only focuses on this version.

As in the case with usual Gibbs sampling, when the second step of the PCG sampler is not in closed form, we can use an Metropolis-Hastings sampler, which yields the Metropolis-Hastings within PCG sampler.

MH within PCG Sampler

For $t = 0, 1, 2, \ldots, T$,

Step 1: Simulate $\upsilon^{(t+1)} \sim p(\upsilon|, Y)$

Step 2: For $i = 1, \cdots, I$, simulate $\omega^{(t+i/I)} \sim \mathcal{K}_{\mathrm{MH}}(\omega|\omega^{(t+(i-1)/I)}, \upsilon^{(t+1)}, Y)$.

The iteration of the Metropolis-Hastings sampler $\mathcal{K}_{\mathrm{MH}}$ in Step 2 aims to eliminate the dependence of its final output, $\omega^{(t+1)}$, on its starting value, $\omega^{(t)}$. Only $\omega^{(t+1)}$ is retained; the intermediate draws, $\omega^{(t+1/I)}, \ldots, \omega^{(t+(I-1)/I)}$ are discarded. The multiple iterations in Step 2 aim to reduce the dependence of $\omega^{(t+1)}$ on $\omega^{(t)}$, so that Step 2 yields the draw of $p(\omega|v^{(t+1)}, Y)$, according to the correct conditional distribution.

We will explore these MCMC algorithms in the context of high-energy astrophysics in the following chapters.

1.5 Principle Component Analysis

In the temperature measuring model of Section 1.1, calibration uncertainty is a one-dimensional quantity. However, in practice, calibration uncertainty could be of high dimensions. For example, the effective area curve (Section 2.3) can be a 1000-dimensional vector, and the photon redistribution matrix (Section 6.1) can be a matrix with 10^6 entries. How to efficiently summarize calibration uncertainty becomes nontrivial. We will use Principle Component Analysis (PCA) to reduce the dimensionality of the calibration uncertainty.

Principal component analysis is a statistical procedure that uses orthogonal transformations to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to (i.e., uncorrelated with) the preceding components. Principal components are guaranteed to be independent if the data set is jointly normally distributed. (More discussion can be found in Jolliffe (2005), Anderson (2003), Ramsay (2006) and Bishop *et al.* (2006))

Consider a data matrix, $X = (x_1^\top, x_2^\top, \ldots, x_n^\top)^\top$, of dimension $n \times p$. Each of the n rows represents one observation vector, x_i, and each of the columns represents one variable vector containing its values of these n observations. Each variable has zero empirical mean, that is, the sample mean of each column is subtracted from that column to make it mean zero. In our astrophysics cases, we may have $p > n$.

To summarize X, consider the singular value decomposition (SVD) of X:

$$X = U\Sigma W^\top$$

where Σ is an n-by-p diagonal matrix with positive entries $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$ on thediagonal, which are known as singular values of X; U is an n-by-n matrix, the columns of which are orthogonal unit vectors of length n called the left singular vectors of X; and W is a p-by-p whose columns are orthogonal unit vectors of length p and called the right singular vectors of X.

Then, each observation x_i can be written as:

$$x_i = \sum_{j=1}^{n} u_{ij}\sigma_j w_j^\top \qquad (1.10)$$

where u_{ij} is the ijth element of U, and $W = (w_1, w_2, \ldots, w_p)$. In other words, each row x_i can be represented by a linear combination of $w_j^\top$'s.

Note that, up to a multiplication constant, $X^\top X$ is the empirical sample covariance matrix, which can also be written as

$$X^\top X = W\Sigma^\top U^\top U\Sigma W^\top = W(\Sigma^\top\Sigma)W^\top$$

where $\Sigma^\top\Sigma$ is $p \times p$ diagonal matrix with positive entries $\sigma_1^2 \geq \sigma_2^2 \geq \ldots \geq \sigma_n^2$ (the other $p - n$ diagonal elements are equal to zero), which is in fact the eigenvalue matrix with each eigenvalue equal to the square of singular values σ_i of X, and the right singular vectors W

of X are equivalent to the eigenvectors of $X^\top X$.

In the view of Equation (1.10), let us consider a linear combination:

$$\tilde{x} = \sum_{j=1}^{n} e_j \sigma_j w_j^\top, \tag{1.11}$$

where e_j's are i.i.d $\mathcal{N}(0,1)$, $j = 1, ..., n$. Then it is easy to show that $E(\tilde{x}) = 0$, and $Cov(\tilde{x}) = X^\top X$. Hence the distribution of $\tilde{x}$ preserves the covariance structure of the data set X.

To reduce the dimensionality, we can modify Equation (1.11) by keeping the first J components in Equation 1.11. That is,

$$\tilde{x} = \sum_{j=1}^{J} e_i \sigma_j w_j^\top.$$

The fraction of the variance that can be explained in these J eigenvectors' directions can be calculated as:

$$f_J = \frac{\sum\limits_{j=1}^{J} \sigma_j^2}{\sum\limits_{j=1}^{n} \sigma_j^2}$$

In practice, this allows us to use a smaller number of components, $J < n$, to approximately capture the covariance structure. When the columns of X are collinear, often only a few principle components are needed.

PCA not only helps reduce the rank of covariance matrix from p to J, but also provides us a simple technique to sample any number of Monte Carlo replicates of x. We can also evaluate the probability density of $\tilde{x}$ by simply multiplying the probability densities of e_j's. These properties will be convenient when we apply PCA to summarize calibration uncertainties in astrophysics (i.e., the effective area curve in Chapter 2 and the photon redistribution matrix in Chapter 6).

1.6 Contributions

Let us outline our contributions to astrophysics briefly. The importance of accounting calibration uncertainty is well known among calibration scientists, but robust principled methods are lacking. We first adopt the Bayesian approach to account for calibration uncertainty of the effective area curve in spectral analysis. In Chapter 3, we illustrate the concept and the advantages of the fully Bayesian method. Built on the work of Lee *et al.* (2011), in Chapter 4, we drastically improve the computational efficiency of the pragmatic Bayesian sampler. Based on this, we develop the main fully Bayesian sampler schemes. Simulation studies and real data analyses are conducted in Chapter 5 to demonstrate the effectiveness of our fully Bayesian method and its potential application in various areas of astrophysics. For example, in Chapter 6, we follow the same procedure as in the case of the effective area curve, to account for calibration uncertainty of the photon redistribution matrix. In other words, our method can be used to handle other calibration products.

Chapter 2

Calibration Uncertainty in High-Energy Astrophysics

In this chapter, we will provide some background of calibration uncertainty in High-Energy Astrophysics. Then, a spectral model including calibration uncertainty will be built up and a Bayesian analysis will be conducted. Here, we will only focus on one source of calibration uncertainty, which is the effective area curve. The PCA summary for effective area curve will be discussed in detail. Lastly, the concept of the pragmatic Bayesian method proposed by Lee *et al.* (2011) will also be introduced in this chapter

2.1 Introduction

In high-energy astrophysics observed data are almost always result from a convolution of photon count data with *instrument calibration products* such as effective area curves, energy redistribution matrices, and point-spread functions. Effective area curves record the detector's sensitivity as a function of energy. When one photon of certain energy reaches the detector, there is a proportion of the detector area that can record this photon. In a statistical view, it can be considered as the probability that one photon with certain energy is detected. Energy redistribution matrices record the probability of the detected photon's spreading into a range of energy bins of the detector, see Section 6.1. Point-spread function describes the response of an imaging system to a point source or point object. A careful specification of these calibration products is critical both for parameter fitting and for properly accounting for the statistical errors of these fits. It is only through instrument calibration that we can transform measured signals into physically meaningful quantities and then interpret data in a meaningful manner. Misspecification of calibration products can lead to bias in the fitted parameters, unreliable statistical errors, and uninterpretable results.

In practice it is well-known that instrumental properties (e.g., the quantum efficiency of a Charge Coupled Device (CCD) detector, point-spread functions, etc.) are measured with error. Unfortunately, typical analyses only account for nominal estimates of calibration products without regard for their errors and/or their possible misspecification. This can seriously degrade fitted parameters and their error bars. In spectral analysis, for example, (Drake *et al.*, 2006) showed that ignoring calibration uncertainty can result error bars that are underestimated by a factor of as much as five, see their Figure 5. We will show that ignoring these errors not only is detrimental to error bars, but can also bias the the fitted values themselves.

Efforts have been made to develop methods that account for calibration uncertainty in high-energy astrophysics and such methods exist both in other areas of astrophysics and in related fields such as particle physics (Heinrich and Lyons, 2007) and observational cosmology (Bridle *et al.*, 2002), see Lee *et al.* (2011) for a review. The nature of the errors in high-energy calibration products including their complex correlations, however, means that such methods are inappropriate and will not provide reliable results. Modern instruments such as the *Chandra X-ray Observatory* are calibrated using data from defined sources obtained either from particularly well-understood astronomical sources or in the lab, and comparing this data with theoretical predictions. These measurements are not typically used directly, but rather they are used to tune sophisticated physics-based computer codes that model the instrument as a whole. These codes can be used to derive both nominal estimates of calibration products as measures of their uncertainty. The calibration products are high-dimensional and exhibit complex and large scale correlation structures among their components. Accounting for this complex uncertainty is further complicated in high-energy astrophysics due to the non-Gaussian nature both of the underlying distribution of source photon counts and of the instrument response. The non-Gaussian character of the data along with the complex correlations in the calibration uncertainty mean that existing methods are not by in large applicable in this setting. The general method of combining measurement and calibration errors in quadrature (e.g., Bevington and Robinson, 1992), for example, assumes Gaussian errors, uncorrelated calibration errors, and that there is a one-to-one relationship between calibration errors and data points. This is not appropriate in the context of the complex correlations exhibited by calibration products in high-energy astrophysics and the enormous consequent variety of possible products. Unfortunately, the choice among these products can effect the final fitted values. Thus, calibration uncertainty must be folded into analyses in a statistically principled manner. As we shall see, doing so can allow the data to inform the choice among the possible calibration products.

To address these complex correlations, Drake *et al.* (2006) suggested a bootstrap-like method that relies on the availability of a large representative sample of possible calibrations products given the calibration uncertainty; we refer to this sample as the *calibration library.* In particular, they propose that a replicate data set be generated for each calibration product in the library and that each replicate data set be fit in the usual way. The variability among the resulting fitted model parameters is then used to estimate the effect of calibration uncertainty on the fitted parameters. While this is a useful method to demonstrate the scale of the effect of calibration uncertainty on error bars for the model parameters, it is not a robust statistical procedure in that generating replicated data sets requires knowledge of typically unknown model parameters. From a practical point of view, the methods reliance on a large calibration library is also problematic, especially considering that calibration products of space-based detectors degrade over time and different calibration products—and hence different calibration libraries—are required for different observations.

Lee *et al.* (2011) proposed to solve these practical problems by first replacing the large calibration library with a low-dimensional model for the calibration uncertainty (derived from a principle component analysis (PCA) of the library) and second embedding the model for calibration uncertainty into a Bayesian procedure that simultaneously fits the model parameters and accounts for calibration uncertainty. By virtue of the the calibration model this strategy effectively embeds the instrument-modeling code as an integral part of the statistical computing techniques. A critical assumption of this approach is that it supposes that the observed photon counts and the calibration product are independent, that is, that the data provide no information for narrowing the calibration uncertainty. An advantage of this independence assumption is that it significantly simplifies the complexity of the necessary computing, in that the algorithm is easy to implement, if not quick to run. For this reason Lee *et al.* (2011) called their approach a *pragmatic Bayesian method.* The independence assumption of the pragmatic Bayesian method also ensures that the choice of calibration product is determined by calibration scientists, calibration experiments, and calibration simulations, rather than the data from a particular observation, which may be viewed as an advantage by some researchers.

The primary objective of this thesis is to remove the independence assumption of the pragmatic Bayesian approach and allow the data to narrow calibration uncertainty. This is a more principled approach from a statistical perspective. If a subset of calibration products are possible according to the pre-specified calibration uncertainty but inconsistent with the observed data, this subset should not play a role in the final analysis. From a statistical perspective, it is the data that should drive a statistical analysis. We call this a *fully Bayesian method* because it is more principled from a statistical point of view. This approach has other advantages. For example, in addition to the calibration uncertainty quantified through idealized experiments, calibration products are subject to errors stemming from differences between these idealized settings and the variety of actual settings in which the products are used. Indeed, suspected systematic errors cannot be fully understood without taking into account the actual data in any particular observation and/or cross-instrument comparisons can be made (e.g., Nevalainen *et al.*, 2010). Our fully Bayesian method allows the data to inform our choice of possible calibration products. In practice, we find that relatively large data sets ($\gg 10^4$ counts) are needed to obtain appreciable power in narrowing calibration uncertainty.

Like the pragmatic Bayesian method, the fully Bayesian approach embeds a model for calibration uncertainty into a larger statistical model. Unlike the pragmatic Bayesian method, however, it then marginalize over calibration uncertainty *while conditioning on the observed data*, whereas the pragmatic method marginalize over calibration uncertainty *without conditioning on the data*. In this regard, the fully Bayesian method is in line with a methods proposed by Bridle *et al.* (2002) and Heinrich and Lyons (2007) for handling systematic errors in cosmology and particle physics, respectively. These proposals, however, use a parameterized form for the systematics under which marginalization can be achieved analytically. While these specific proposals are not applicable in our setting, they share our emphasis on the general principle of building a joint model that incorporates all sources of uncertainty and then marginalizing over nuisance parameters while conditioning on the observed data.

While the statistical framework that we present is quite general, for clarity we focus on high-energy spectral analysis with uncertainty in the effective area curve for the remainder of this thesis. In the remainder of this section, we outline the necessary background on Bayesian spectral analysis, Bayesian model fitting using Markov chain Monte Carlo, the

PCA-based calibration model proposed by Lee *et al.* (2011), and their pragmatic Bayesian method. In Chapter 3, we lay out our fully Bayesian method and illustrate its advantage over both the typical strategy of ignoring calibration uncertainty and the pragmatic Bayesain method using a simple numerical illustration. The fully Bayesian methods requires us to simultaneously fit the model parameters and the effective area curve while accounting for pre-specified calibration uncertainty. Section 4.2 outlines how we leverage the `PyBLoCXS` module in Sherpa along with the pragmatic Bayesian method to derive a sophisticated and computationaly efficient algorithm for the fully Bayesian method. Section 4.3.2 describes the main fully Bayesian algorithm that is applied for the following numerical analyses. In Section 5.1 and 5.2 we validate our proposal using a set of simulation studies that include a comprehensive frequency evaluation. We find striking improvement when large-count spectra ($\gg 10^4$ counts) generated with an effective area curve that is consistent with the pre-specified calibration uncertainty are fit with a misspecified default curve. In Section 5.3, 5.4 and 5.5 we illustrate how the fully Bayesian method works in practice using several data sets obtained with the Chandra telescope: a collection of quasi-stellar objects (QSO's) observed near the aimpoint of ACIS-S (ACIS stands for Advanced CCD Imaging Spectrometer) and described with absorbed power-law models; a bright O system at a large off-axis location on ACIS-S2 and modeled as absorbed multi-thermal spectra; and co-added long-duration grating observation of an isolated neutron star modeled as a blackbody spectrum. In Chapter 6, we discuss the calibration uncertainty in photon redistribution matrix and prove that the fully Bayesian algorithm can also applied to redistribution matrix uncertainty. Finally discussion appears in Chapter 7 and technical details about `PyBLoCXS` given in an Appendix A. A glossary of the symbols we use in the following chapters is given in Table 2.1.

Table 2.1: Glossary of symbols used in the Chapter 2-6

A	effective area (ARF) curve
A_0	the default effective area curve.
A_0^*	the observation specific effective area curve.
A_l	effective area curve l in the calibration library
$\bar{A}$	the average of the effective area curves in the calibration library
A^{prop}	a proposed effective area curve in an MH sampler
$A_{\text{fB}}^{(t+1)}$	an effective area curve simulated with the fully Bayesian sampler
$A_{\text{pB}}^{(t+1)}$	an effective area curve simulated with the pragmatic Bayesian sampler
$\mathcal{A}$	a set of effective areas, the calibration library
B	the between imputation (or systematic) variance of $\hat{\theta}$.
$\mathcal{CI}$	a confidence interval
E	energy of incident photon
E^*	energy channel at which the detector registers the incident photon
e	the low-dimensional PCA representation of A, see Equation 2.4
e^{prop}	value of e proposed in an MH sampler
$e_{\text{fB}}^{(t+1)}$	value of e simulated with the pragmatic Fully sampler
$e_{\text{pB}}^{(t+1)}$	value of e simulated with the pragmatic Bayesian sampler
Q	generic proposal distribution in a MH sampler
I	number of inner iterations in `pyBLoCXS`, typically 10
i	inner iteration number or index
$\mathcal{I}$	information obtained prior to the data, for example by calibration scientists
J	number of components used in PCA analysis, here 8
j	principal component number or index
k	row indicator of photon redistribution matrix
K	column dimension of photon redistribution matrix
$\mathcal{K}$	an MCMC kernel
$\mathcal{K}_{\text{pyB}}$	the MCMC kernel used in `PyBLoCKS`
L	number of replicate effective area curves in calibration library
L	replicate effective area number or index
$L(\cdot)$	the likelihood function
M	number of replicated draws of θ per draw of A in Iterated MH within PCG sampler
n	row indicator of photon redistribution matrix
N	row dimension of photon redistribution matrix
$p_{\text{pB}}, p_{\text{fB}}$	pragmatica and fully Bayesian posterior distribution
R	photon redistribution matrix (RMF)
$\mathcal{R}$	photon redistribution matrix library
R^b	photon redistribution matrix with large value entires, along the right diagonal line
R^s	$R - R_b$
$\tilde{R}$	vectorized R

Table 2.2: Glossary of symbols used in the Chapter 2-6 - Continued

r_j^2	eigenvalue or PC coefficient of component l in the PCA representation
T	number of MCMC iterations
t	main MCMC iteration number or index
$^{(t+\ldots)}$	the superscript indicates the running index of random draws
u	a uniformly distributed random number between zero and one
v_j	eigen- or feature-vector for component l in the PCA representation
W	the within one imputation variance of $\hat{\theta}$.
Y	data, typically used here as counts spectra in detector PI bins
$\hat{\alpha}$	correlation used to validate the choice of I in the MH within PCG sampler
α	the acceptance probability in an MH sampler
π	a generic prior or posterior distribution
ρ	lag one autocorrelation of the Monte Carlo samples from standard method
τ	computational time
θ	spectral model parameter
$\hat{\theta}$	estimate of θ
θ^{prop}	a value of θ proposed in an MH sampler
$\theta_{\mathrm{fB}}^{(t+\ldots)}$	a value of θ simulated with the fully Bayesian sampler
$\theta_{\mathrm{pB}}^{(t+\ldots)}$	a value of θ simulated with the pragmatic Bayesian sampler
ψ	generic notation for unknown quantities in analysis, e.g, $\psi = \theta$ or $\psi = (A, \theta)$
ψ^{prop}	a value of ψ proposed in an MH sampler
$\hat{\sigma}_{\mathrm{std}}$, $\hat{\sigma}_{\mathrm{pB}}$, $\hat{\sigma}_{\mathrm{fB}}$	error bars under fixed effective area, pragmatic, and fully Bayesian methods
Σ	variance-covariance matrix of $\hat{\theta}$
Ψ	normal approximation parameters from R
$\ominus$	library of Ψ

2.2 Bayesian Spectral Analysis

The observed photon count in energy channel E^* is modeled by a Poisson distribution,

$$Y(E^*) \sim Pois\left(\sum_E \Lambda(E;\theta)A(E)R(E^*;E) + B(E^*)\right), \tag{2.1}$$

where $\Lambda(E;\theta)$ is the source spectral intensity in energy bin E, θ is the spectral (source) parameter of primary interest, $A(E)$ is the effective area curve in energy bin E, $R(E^*;E)$ is the energy redistribution matrix of detector, and $B(E^*)$ is the background intensity in channel E^*. The photon counts in each channel, $Y(E^*)$ are independent Poisson variables. For simplicity, we represent the vector of observed photon counts by $Y = \{Y(E^*)\}$, the effective area by $A = \{A(E)\}$, and the photon redistribution matrix by $R = \{R(E^*;E)\}$. In our numerical analyses, we consider an energy range from 0.3keV to 7keV, which is divided evenly into approximately one thousand bins.

To fit the source parameters, θ, given the observed photon counts, Y while accounting for calibration uncertainty, we adopt the Bayesian method. In particular, suppose ψ represents the unknown quantities, and $\mathcal{I}$ is the information we have before seeing the data. In this thesis, we treat θ and A as unknown, so $\psi = (\theta, A)$ while R is part of $\mathcal{I}$. This is in contrast to a standard analysis, where A is treated as known and is a part of $\mathcal{I}$ rather than of ψ.

In this thesis, we focus on Bayesian statistical methods that allow us to simultaneously learn about θ and A. That is, we treat A as an unknown quantity and allow its uncertainty to affect the fit and error bars of θ. Similar methods would be employed to account for uncertainty in R in Chapter 6.

Substituting $\psi = (\theta, A)$ into Equation (1.2) and assuming the priors for θ and A are independent, we can write the posterior distribution.

$$p(\theta, A|Y, \mathcal{I}) \propto L(Y|\theta, A, \mathcal{I})\pi(\theta|\mathcal{I})\pi(A|\mathcal{I}), \tag{2.2}$$

we typically use a defuse prior distribution on θ to reflect lack of knowledge on θ and an informative prior on A representing information contained in $\mathcal{I}$ that is obtained from calibration studies. In the remainder of the thesis, we typically omit $\mathcal{I}$ in our notation.

Unlike the simple temperature measuring example, the posterior distribution of the Bayesian spectral method has no analytical solutions. We need to apply MCMC algorithms to obtain numerical results. This will be the main target of this thesis. van Dyk *et al.* (2001) and Park *et al.* (2008) detail the use of Bayesian MCMC for spectral analysis and give general guidance on the use of MCMC algorithms in high-energy astrophysics.

While our primary goal is to consider methods for joint inference for θ and A using Equation (2.2), we also compare such methods with the standard approach that treats A as fixed and known. For clarity we refer this approach as the *standard method.* For example, in a Bayesian analysis, the standard method involves estimating θ using its posterior distribution given the observation, Y, and the nominal effective area curve associated with this observation, A_0^*. As a result,

$$p_{\text{std}}(\theta|Y, A_0^*) \propto L(Y|\theta, A_0^*)p(\theta|A_0^*) \tag{2.3}$$

where $p(\theta|A_0^*)$ represents the prior distribution of θ, given A_0^*. Usually we use prior distribution for θ that is independent of A, $p(\theta|A_0^*) = \pi(\theta)$. This approach fixes $A = A_0^*$, and it does not incorporate the calibration uncertainty in A. Lee *et al.* (2011) et al illustrates that the standard method can lead to misleading estimates of θ and can significantly underestimate the error bars associated with these estimates. Nevertheless, because this is the standard approach in practice, we treat it as a baseline in our numerical comparisons. Here the subscript std indicates that this posterior distribution is from standard method.

A Sherpa module, `PyBLoCXS` (Bayesian Low Count X-ray Spectral analysis in Python) adopted from the method of van Dyk *et al.* (2001), is designed to use MCMC algorithms for spectral analysis in the Sherpa environment. It provides us a convenient and efficient MCMC sampler for simulating spectral source parameters θ from the posterior distribution $p(\theta|Y, A)$, (A is assumed to be fixed and known in `PyBLoCXS`). This module is designed to conduct the standard method. More detail about `PyBLoCXS` can be found in Appendix A. In this thesis, we use the `PyBLoCXS` algorithm to sample $p(\theta|A, Y)$ and use the notation $\mathcal{K}_{\text{pyB}}$ to refer to the pyBLoCXS kernel. Specifically, `PyBLoCXS` will be incorporated as a component of an overall algorithm that simulates the full posterior distribution, $p(\theta, A|Y)$. To do this, we must quantify what we know about A into a prior distribution, $\pi(A)$. This is the task of Section 2.3

2.3 Quantifying Calibration Uncertainty

The specification of the posterior distribution in Equation 2.2 requires that we formulate a prior distribution on A to encapsulate the calibration uncertainty. Although they were not working in a Bayesian setting, Drake *et al.* (2006) suggested using a set of effective area curves that represent the range of plausible curves and are constructed by calibration scientists to represent calibration uncertainty. Generally speaking, we use the term *calibration product* to refer to an effective area curve, photon redistribution matrix, point spread function, etc. and the term *calibration sample* to refer to a set of such products that summarize calibration uncertainty. Drake *et al.* (2006), for example, simulated a sample of ACIS effective area curves, representative of the possible true curve given our uncertainty. This was accomplished by explicitly including uncertainties in each of the detector's subsystems (UV/ion shield transmittance, CDD quantum efficiency, and the telescope mirror reflectivity). We follow Lee *et al.* (2011) and use the ACIS calibration sample of Drake *et al.* (2006) in our numerical studies. It consists of $L = 1000$ simulated effective area curves, $\mathcal{A} = \{A_1, A_2, ..., A_L\}$. We define $\bar{A}$ to be the arithmetic mean of the calibration sample and let A_0 denote the default effective area curve associated with the sample. $\bar{A}$ and A_0 are typically similar, but may not be equal.

In practice, the calibration sample must be large enough to fully represent the uncertainty in high-dimensional calibration products. To summarize this sample into a concise and usable form, Lee *et al.* (2011) proposed conducting a Principle Component Analysis (PCA) on the mean-subtracted calibration sample, $\{A_1 - \bar{A}, ..., A_L - \bar{A}\}$. PCA is a mathematical procedure that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of linearly uncorrelated variables called principal components. According to Lee *et al.* (2011), in the case of ACIS calibration sample, approximately 20 principal components (out of 1000) account for 99% of the variability in the sample.

Following Lee *et al.* (2011), we conduct a Bayesian analysis that treats the true effective area curve, along with the spectral parameters as unknown. We use the PCA summary of the calibration sample to formulate the prior distribution for A, $\pi(A)$. In particular, we assume that under the prior distribution on A,

$$A(e) = \bar{A} + (A_0^* - A_0) + \sum_{j=1}^{J} e_j r_j v_j, \tag{2.4}$$

where A_0^* is the user-generated observation-specific effective area, r_j^2 and v_j are the first J principle component eigenvalues and eigenvectors, respectively. And e_j is independent standard normal deviations. Since $\bar{A} \approx A_0$, we can view Equation (2.4) as starting with the user-speficied effective are, A_0^*, and adding the random term $\sum_{j=1}^{J} e_j r_j v_j$ to account for uncertainty; $\bar{A} - A_0$ adjust for the necessary mean-subtraction of $\mathcal{A}$ when conducting PCA.

To simulate replicated effective area curves under the prior distribution given in Equation (2.4), we only need to simulate J independent standard normal deviations, $(e_1, ..., e_J)$, and evaluate Equation (2.4). We treat $A(e)$ as the generic notation for the effective area curve and contine to simply use A when its explicit dependence on e is not pertinent.

In effect, we are assuming that the uncertainty in the effective area curve can be described by a 1078 dimensional multivariate normal distribution. The similarity of the effective area curves in $\mathcal{A}$ means that most of the correlations among the components of this distribution are very strong (i.e. near 1). The PCA representation in Equation (2.4) increases the strongest, and the vast majority (1078-J), of these correlation to one and thereby effectively reduces the rank of prior variance-covariance matrix of $\mathcal{A}$. Equation (2.4) also stipulates that the distributions associated with calibration uncertainty for observation-specific effective area curves differ only in their means and that they all have the same variance. This means we can use the variance of Drake et al.'s simulated calibration sample to represent the variance of any observation-specific effective area. In practice, this procedure avoids generating a calibration sample for each observation while still allowing us to account for uncertainty in an effective and efficient manner.

Figure 2.1 illustrates the performance of PCA in summarizing the structure of individual effective area curve in $\mathcal{A}$. It shows that when we use $J = 8$ principle components, the reconstructed effective area curve nicely capture the structures of the original $A \in \mathcal{A}$. This means that we can reduce the dimension of effective area curve to 8 when using PCA. We use $J = 8$ in all the numerical studies.

This PCA-based emulation of the uncertainty in A is critical both for Lee *et al.* (2011)'s Pragmatic Bayesian Method and our proposed Fully Bayesian Method. It not only provides a simple way to incorporate the uncertainty in the effective area curve, but also allows us to evaluate $\pi(A)$ and using Equation (2.2), $p(A, \theta|Y)$. We will need these quantities to implement our MCMC sampler.

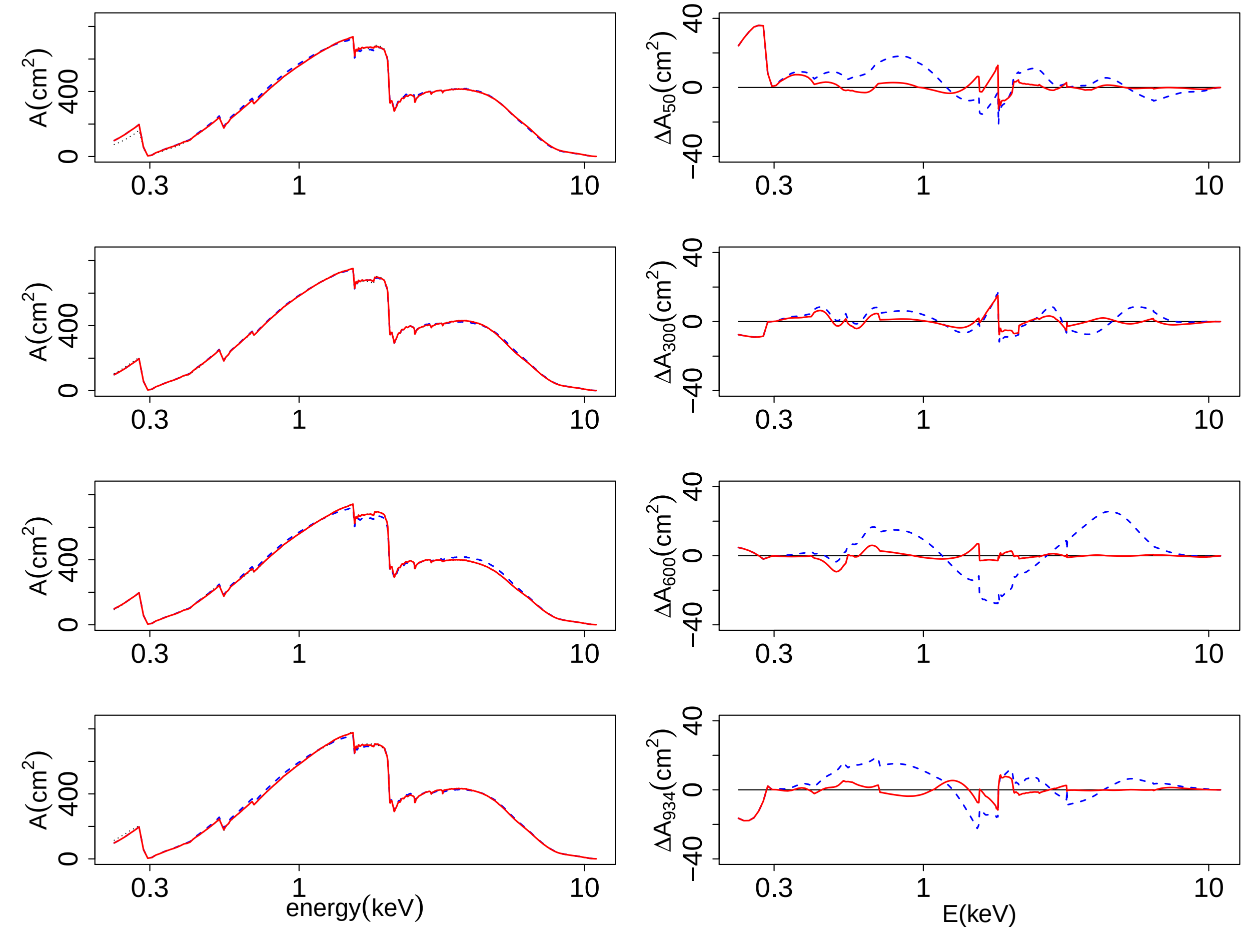

Figure 2.1: The PCA representation of several effective area curves from the calibration library. The left column plots four randomly selected $A_l \in \mathcal{A}$, one in each row, along with their PCA representation, $A_l(e)$, for two values of J. The original curves are plotted in dot-dashed black and the PCA representations are plotted as dashed blue and solid red for $J = 1$ and 8, respectively. The right column is constructed in the same manner, but subtracting off each of the four original effective area curves $\Delta A_l = A_l(e) - A_l$. Although $A_l(e)$ deviates from A_l, even with $J = 8$, the left column shows that the scale of this deviation is quite small and that overall using $J = 8$ concisely captures the structure of each of the effective area curves.

2.4 A Pragmatic Bayesian Method

As noted above (see Equation (2.3)), standard analyses assume that the effective area is fixed. That is, the parameters are estimated conditional on A_0^*. Here we aim to eliminate this conditioning. Mathematically, this involves treating A as unknown rather that conditioning on its value, and expressing Equation (2.2) as

$$p_{\mathrm{fB}}(\theta, A|Y) = p(\theta|A, Y)p(A|Y). \tag{2.5}$$

Because the left-hand side of Equation (2.5) does not condition on the effective area, but rather treat it as an unknown quantity, this model effectively allows us to account for the uncertainty in A. Here the subscript fB indicates that this is fully Bayesian posterior distribution.

Lee *et al.* (2011) made the simplified "pragmatic" assumption that $p(A|Y) = \pi(A)$, where $\pi(A)$ represents the prior distribution for A (e.g. Equation (2.4)). This assumption says that the observed photon counts and the effective area curve are independent; in other words, the data provides no information for narrowing the uncertainty in the choice of effective area curve. Under this assumption, the posterior distribution of θ and A can be written as

$$p_{\mathrm{pB}}(\theta, A|Y) = p(\theta|A, Y)p(A|Y) = p(\theta|A, Y)\pi(A) \tag{2.6a}$$

where $p(\theta|A, Y)$ is given in Equation (2.3) with A_0^* replaced by the generic A. We use the subscript pB in Equation 2.6a to emphasize that this is the posterior distribution under the pragmatic assumption of Lee *et al.* (2011).

Under the model in Equation (2.6a), inference for θ is based on its marginal posterior distribution,

$$p_{\mathrm{pB}}(\theta|Y) \quad = \quad \int p(\theta|A,Y)\pi(A)dA \tag{2.6b}$$

The pragmatic Bayesian method accounts for calibration uncertainty in a conservative manner. The assumption $p(A \mid Y) = \pi(A)$ ignores information in the data that may contribute to the uncertainty in A and hence in θ. We now consider methods that allow Y to narrow the uncertainty of A.

Chapter 3

The Fully Bayesian Solution

In this chapter, we will provide the fully Bayesian solution, which avoids the "pragmatic" assumption in Equation (2.6a). The fully Bayesian method has the unique advantage to narrow the choice of effective area curve, by selecting the effective area curves that are most consistent to the data. The theory and the advantage of the fully Bayesian will be discussed in this chapter through one simulated example.

3.1 Motivation and Theory

To avoid the "pragmatic" assumption in Equation (2.6a), we employ a fully Bayesian approach that makes inference on the full posterior distribution $p_{\mathrm{fB}}(\theta, A|Y)$ in Equation (2.5). This can have practiced consequences, because we find that some effective area curves in the calibration sample can be inconsistent with the observed data, especially with moderate to large data sets. That is, the data themselves not only carry the information about source parameters θ, but also give us evidence as to the choice of effective area curve.

Under the fully Bayesian posterior distribution, the marginal distribution of θ given in Equation (2.6b) is

$$p_{\mathrm{fB}}(\theta|Y) \quad = \quad \int p(\theta|A,Y)p(A|Y)dA$$

which can be weighted version of Equation (2.6b) via

$$\begin{aligned} p_{\mathrm{fB}}(\theta|Y) \quad &= \quad \int p(\theta|A,Y)\frac{p(A|Y)}{\pi(A)}\pi(A)dA \\ &\approx \quad \frac{1}{T}\sum_{t=1}^{n} p(\theta|A^{(t)},Y)\frac{p(A^{(t)}|Y)}{\pi(A^{(t)})}, \end{aligned} \tag{3.1}$$

where $A^{(t)} \sim \pi(A)$. In principle, we could implement the fully Bayesian method by obtaining a sample $\{A^{(t)}, t = 1, ..., T\}$ from $\pi(A)$ and evaluating Equation (3.1). (This technique is known as importance sampling.) Unfortunately the calculation for $p(A|Y) = \int p(\theta, A|Y)d\theta$ can be extremely challenging. This means we cannot perform the fully Bayesian analysis by simply sampling effective area curves from the prior $\pi(A)$ and refitting the model using the standard method for each sampled A, as is done in the pragmatic Bayesian method. Therefore, alternative computational techniques therefore are needed. We outline our strategy in detail in Section 4.3.2 in detail.

An important difference between the fully Bayesian and the pragmatic Bayesian strategies is that the fully Bayesian method leverages the data to learn about A. Specifically, the posterior distribution of A under the pragmatic method, is

$$p_{\mathrm{pB}}(A \mid Y) = \int p_{\mathrm{pB}}(\theta, A \mid Y)\ d\theta = \int p(\theta \mid A, Y)\pi(A)\ d\theta = \pi(A),$$

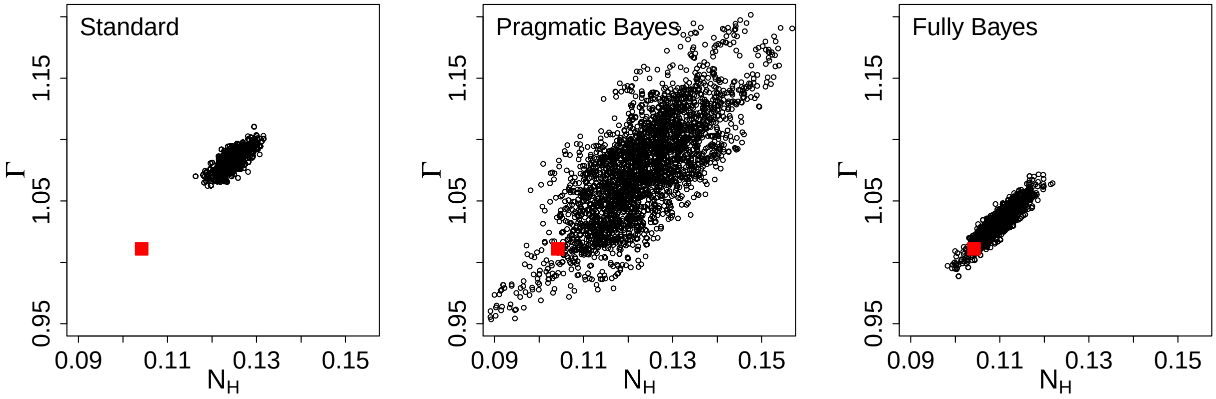

Figure 3.1: Comparison of the standard (left), pragmatic Bayesian (middle), and fully Bayesian (right) methods. Each panel compares a Monte Carlo posterior sample of values of $\theta = (\Gamma, N_{\rm H})$ to its true value, marked with a red square. When fit with an incorrect effective area curve, the standard method can result in misleading estimates of θ (see left panel). The pragmatic Bayesian method, on the other hand, averages over all a priori possible effective area curves and significantly enlarges the posterior variance for θ (see middle panel). Although the centers of the posterior distributions under the standard and pragmatic Bayesian methods are similar the larger error bars computed with the latter allows them to include the true values of θ. Finally the posterior distribution under the fully Bayesian method shifts toward the true value of θ, allowing it to cover the true value while maintaining relatively modest error bars (see right panel).

which is equal to the prior distribution of A. In contrast, under the fully Bayesian posterior,

$$p_{\rm fB}(A \mid Y) = \int p_{\rm fB}(\theta, A \mid Y)\ d\theta = \int p(\theta \mid A, Y)\ p(A \mid Y)\ d\theta = p(A \mid Y)$$

can be used to learn what effective area curves are more or less consistent with the observed data.

3.2 The Advantage of the Fully Bayesian Analysis

To illustrate the advantage of the fully Bayesian method over the pragmatic Bayesian method, we compared their performance in a simulation study. In Section 5.1 we reproduce part of the simulation study of Lee *et al.* (2011) but this time we include the fully

Bayesian method. Here we give detailed results under one setting of this simulation study to illustrate the potential of the fully Bayesian method before discussing the necessary computational techniques. In particular, we simulate an absorbed power-law source model with three parameters (power-law index Γ, absorption column density $N_{\rm H}$, and normalization) using the `fake_pha` routine in Sherpa. The data set was simulated without background contamination using the XSPEC model `wabs*powerlaw` and a default (RMF) for ACIS-S. Lee *et al.* (2011) replicated the simulation under eight settings that varied the hardness of the power law, the choice of effective area, and the nominal counts, details appear in Section 5.1. We consider the energy range from 0.3keV to 7keV, which is divided evenly into approximately one thousand bins. Here we give detailed results for Simulation II which set $\Gamma = 1$, $N_{\rm H} = 10^{21}{\rm cm}^{-2}$, generated 10^5 counts, and used an off-center effective area curve from the calibration library of (Drake *et al.*, 2006). (We use the same off-center effective area curve as Lee *et al.* (2011) and label it $A_{\rm ext}$; it is number 934 in the calibration library.)

Figure 3.1 plots a Monte Carlo sample from the posterior distribution under the standard method, $p(\theta \mid A_0, Y)$, the pragmatic Bayesian method $p_{\rm pB}(\theta|Y)$, and the full Bayesian method, $p_{\rm fB}(\theta \mid Y)$, where A_0 is the default under the calibration library of Drake *et al.* (2006) and here $\theta = (\Gamma, N_{\rm H})$. The red square in each panel gives the true value of θ. The MCMC samplers used in Figure 3.1 are described in Chapter 4. Although the error bars computed with the standard method are the smallest, in this simulation the method misses the true true value of θ. Thus, the results of the standard method are precise but incorrect. This is not unusual when the default effective area curve, A_0, is misspecified, as it is in this case because the data were generated under a different curve—one that is nonetheless plausible given the calibration uncertainty. The pragmatic method accounts for calibration uncertainty by averaging over all *a priori* possible effective area curves, resulting in much larger error bars that capture the true value of θ – the method is imprecise but correct. As pointed out by Lee *et al.* (2011), this is a clear advantage over the standard method. Finally, the fully Bayesian method accounts for calibration uncertainty by averaging over those effective area curves that are consistent with the observed data. The resulting error bars are only slightly larger than those produced with a fixed effective area curve, but the fitted values for θ have shifted enough that the error bars still capture the true value. This example clearly illustrates the benefits of the fully Bayesian method: we can characterize the performance of the standard method as precise but wrong, that of the pragmatic method as imprecise but correct and that of the fully Bayesian method as both precise and correct. We conduct additional numerical study in Sections 5.1 and 5.2 to illustrate when we can expect the fully Bayesian method to outperform the standard method and pragmatic Bayesian method.

Chapter 4

Statistical Computation

In previous chapter, we found that the fully Bayesian method outperforms the standard method or the pragmatic Bayesian method. In this chapter, we outline our computational strategy for fitting the fully Bayesian method. Firstly, we describe one of the main algorithms for the pragmatic Bayesian method proposed by Lee *et al.* (2011), MH within Partial Collapsed Gibbs Sampler. Based on this algorithm, we develop a more efficient pragmatic Bayesian algorithm, called Iterated MH within Partial Collapsed Gibbs Sampler. It highly leverages each update of the effective area curve, by keeping multiple draws of spectral parameters corresponding to one effective area curve. Then, the basic HM within Gibbs Sampler for the fully Bayesian is proposed, which turns out to be not applicable in reality. Lastly, the pragmatic proposal sampler for the fully Bayesian method is developed, by using the samples from the pragmatic Bayesian as the proposal distribution. The advantages of this sampler are detailed in this chapter.

4.1 The Original Pragmatic Bayesian Sampler

To fit the pragmatic Bayesian model, Lee *et al.* (2011) proposed a number of algorithms including a Metropolis Hastings (MH) with Partially Collapsed Gibbs (PCG) sampler. This involves simulating a sequence of effective area curves from $\pi(A)$ and running `PyBLoCXS` for each simulated curve. Specifically, in their Section 4.2.2, they proposed

MH within PCG Sampler of Lee *et al.* (2011)

For $t = 0, 1, 2, \dots, T$,

Step 1: Simulate $e_j \sim \mathcal{N}(0, 1)$ for $j = 1, \dots, J$ and set

$$e_{\text{pB}}^{(t+1)} = (e_1^{(t+1)}, \dots, e_J^{(t+1)}) \text{ and } A_{\text{pB}}^{(t+1)} = \bar{A} + (A_0^\star - A_0) + \textstyle\sum_{j=1}^J e_j^{(t+1)} r_j v_j.$$

Step 2: For $i = 1, \cdots, I$, simulate $\theta_{\text{pB}}^{(t+i/I)} \sim \mathcal{K}_{\text{pyB}}(\theta \mid \theta^{(t+(i-1)/I)}; Y, A^{(t+1)})$.

The iteration of `PyBLoCXS` in Step 2 aims to eliminate the dependence of its final output, $\theta_{\text{pB}}^{(t+1)}$, on its starting value, $\theta_{\text{pB}}^{(t)}$. Only $\theta_{\text{pB}}^{(t+1)}$ is retained; the intermediate draws, $\theta_{\text{pB}}^{(t+1/I)}, \dots, \theta_{\text{pB}}^{(t+(I-1)/I)}$ are discarded. This is necessary because $A_{\text{pB}}^{(t+1)}$ is simulated from its marginal distribution in Step 1 and is independent of $\theta_{\text{pB}}^{(t)}$. The simulated value $\theta_{\text{pB}}^{(t+1)}$ should be correlated with $A_{\text{pB}}^{(t+1)}$, insofar as A is informative for θ and thus A and θ are correlated under $p_{\text{pB}}(\theta, A \mid Y)$. Likewise, because $A_{\text{pB}}^{(t)}$ and $A_{\text{pB}}^{(t+1)}$ are independent, $\theta_{\text{pB}}^{(t)}$ and $\theta_{\text{pB}}^{(t+1)}$ should also be independent. The number of inner iterations, I, is determined by examining the empirical autocorrelation function of $\mathcal{K}_{\text{pyB}}$ for a given data analysis. Its value should be set to ensure that $\theta_{\text{pB}}^{(t)}$ and $\theta_{\text{pB}}^{(t+1)}$ are independent, see Lee *et al.* (2011) and van Dyk and Jiao (2014) for discussion.

While Lee *et al.* (2011) showed that this MH within PCG sampler effectively delivers fitted values and error bars that account for calibration uncertainty under $p_{\text{pB}}(\theta, A \mid Y)$, it can be very slow in terms of its required computational time. This is owing to the inner workings of `PyBLoCXS` which approximates $p(\theta \mid A, Y)$ by fitting the spectral model via maximum likelihood (i.e., using the Cash statistic) in Sherpa. This is of little consequence when `PyBLoCXS` is used with fixed A, because the Sherpa fit only needs to be preformed once. In the context of the MH within PCG Sampler, however, `PyBLoCXS` must be reinitiated with a new Sherpa fit at every iteration because $A_{\mathrm{pB}}^{(t+1)}$ is updated at every iteration. Because each Sherpa fit requires about 6-8 seconds of CPU time, a run of 3000 MCMC iterations spends 5-7 hours on Sherpa fits.

4.2 Improving the Pragmatic Bayesian Sampler

4.2.1 An Iterated Pragmatic Bayesian Sampler

The Pragmatic Bayesian sampler only generates one draw of θ for each effective-area specific Sherpa fit. Although this results in (nearly) independent simulates of θ, the computation is expensive. We propose an alternate implementation of the MH within PCG Sampler that is dramatically faster. In particular, it fully leverages each of the effective-area specific Sherpa fits by continuing to iterate as in Step 2 after the dependence on $\theta_{\mathrm{pB}}^{(t)}$ has worn off in order to obtain several simulated values of θ per iteration. Specifically, we propose

Iterated Pragmatic Bayesian Sampler:

For $t = 0, 1, ..., T$

Step 1: Simulate $e_j^{(tM+1)} \sim \mathcal{N}(0,1)$ for $j = 1, ..., J$ and set $e_{\mathrm{pB}}^{(tM+1)} = (e_1^{(tM+1)}, ..., e_J^{(tM+1)})$ and $A_{\mathrm{pB}}^{(tM+1)} = \bar{A} + (A_0^* - A_0) + \sum_{j=1}^{J} e_j^{(tM+1)} r_j v_j$.

Step 2: For $i = 1, ..., I$, simulate $\theta^{(Mt+i/I)} \sim \mathcal{K}_{\mathrm{pyB}}(\theta | \theta^{(Mt+(i-1)/I)}; Y, A^{(t+1)})$.

Step 3: For $i = 2, ..., M$, simulate $\theta_{\mathrm{pB}}^{(Mt+i)} \sim \mathcal{K}_{\mathrm{pyB}}(\theta|\theta^{(Mt+i-1)}; Y, A^{(t+1)})$

Just as with original Pragmatic Bayesian sampler, we discard $I - 1$ sub-iterations in Step 2 to ensure that draws corresponding to different effective area curve are independent of each other. This ensures that stationary distribution of the chain is the target posterior distribution.

At iteration t, we only retain $e_{\mathrm{pB}}^{(tM+1)}$, $A_{\mathrm{pB}}^{(tM+1)}$, and $\theta_{\mathrm{pB}}^{(tM+1)}, \cdots, \theta_{\mathrm{pB}}^{(Mt+M)}$ and discard the intermediate draws of θ obtained in Step 2. Notice that this strategy results in M times more simulated values of θ than those of A. The advantage is that the extra $T(M-1)$ simulations of θ require relatively little computational time. The disadvantage is that these simulated values are correlated whereas the simulated values of the MH within PCG sampler of Lee *et al.* (2011) are essentially independent.

To illustrate the computational advantage of the iterated pragmatic Bayesian sampler, we compared its performance with the original sampler in a simulation study that is described in Section 3.2. The left and right columns of Figure 4.1 show the performance of the MH within PCG Sampler of Lee *et al.* (2011) and our Iterated Sampler (run with $M = \mathbf{10}$), respectively. The first row is a time-series plot of the first 300 consecutive draws of $\Gamma^{(t)}$ obtained with the two samplers. The blocky nature of the draws obtained with the Iterated Sampler is a result of its infrequent updating of A; each block is drawn with a common effective area curve. The second row presents the same time-series plots, but as a function of CPU time (in second) rather than iteration number. The relative speed of the Iterated Sampler is apparent. The final row shows scatterplots of the of the simulated values of $(N_{\mathrm{H}}^{(t)}, \Gamma^{(t)})$ obtained in about one hour. Again the advantage of the Iterated Sampler is clear, despite the blocky nature of its simulated values.

Running the Iterated Sampler requires the values of both I and M, that is, the number of initial simulated values of θ to discard $(I-1)$ at each iteration and the number of additional

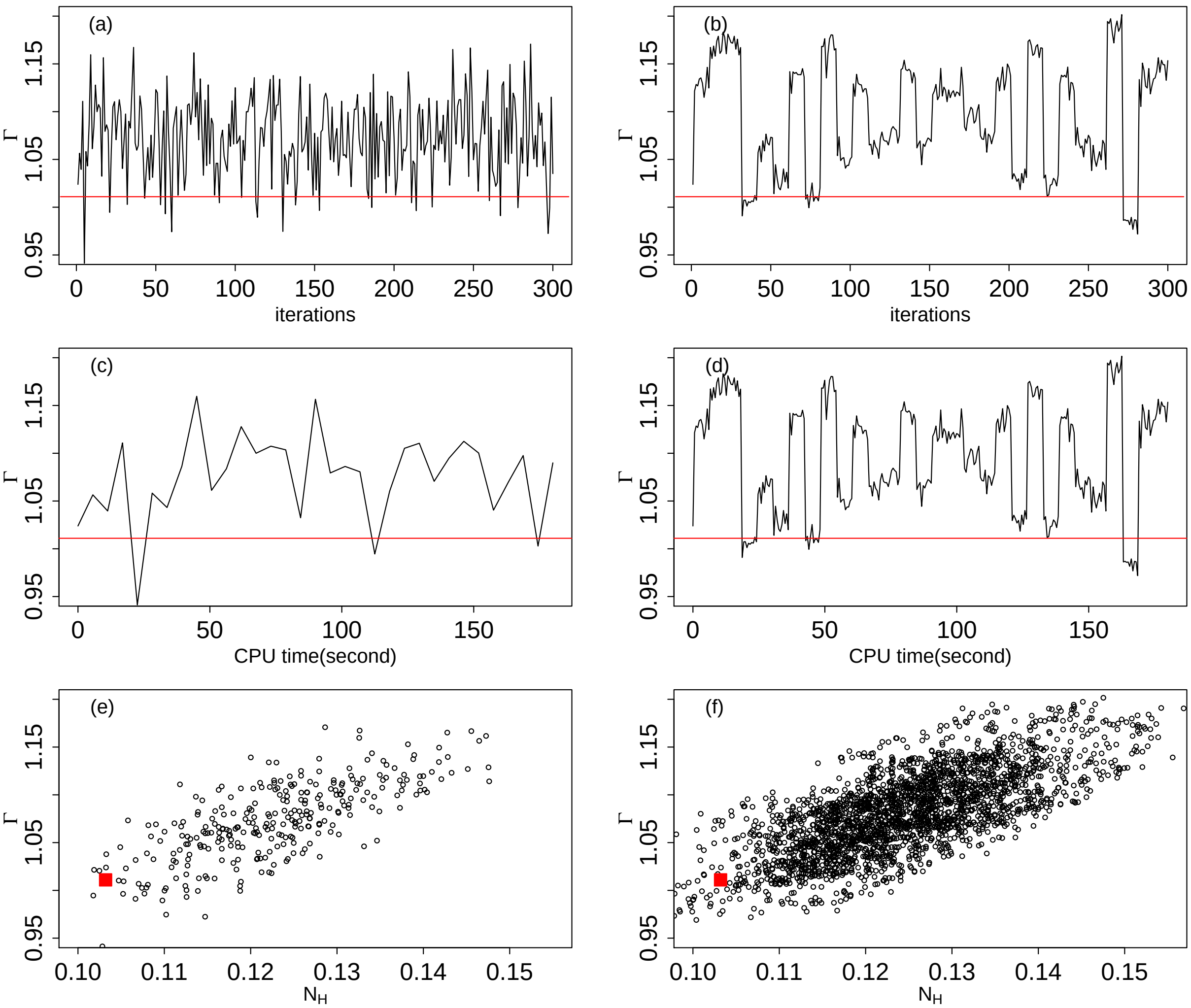

Figure 4.1: The improved speed of the iterated MH within PCG sampler. Using the simulation study described in Section 3.2, panels (a)-(b) plot $\Gamma^{(t)}$ as a function of iteration number using the sampler of Lee *et al.* (2011) and the intreated MH within PCG sampler, respectively. Panels (c)-(d) plot the same, but as a function of CPU time rather than of iteration. The relative speed of the iterated method is apparent. Panels (e)-(f) are scatterplots of $(\Gamma^{(t)}, N_{\rm H}^{(t)})$ obtained in the first hour of the two samplers. The red lines in panels (a)-(d) and squares in panels (e)-(f) indicate the true parameter values. Despite the blocky nature of the chains in panels (b) and (d), the advantage of the iterated sampler in terms of computational speed is clear.

extra simulated values to keep (M). The initial iteration in Step 2 is designed to mitigate the dependence of the simulated values on the input value of θ at each iteration. To measure this effect, we can compute the correlation of $\theta^{(Mt)}$ and $\theta^{(Mt+1)}$,

$$\hat{\alpha} \approx \frac{\sum_{t=1}^{T}(\theta^{(Mt)} - \bar{\theta}_{\rm pB})(\theta^{(Mt+1)} - \bar{\theta}_{\rm pB})}{\sum_{t=1}^{T}(\theta^{(Mt)} - \hat{\theta}_{\rm pB})^2}, \tag{4.1}$$

where $\bar{\theta}_{\rm pB} = \frac{1}{TM}\sum_{t=1}^{TM}\theta_{\rm pB}^{(t)}$. If $T\hat{\alpha}^2/(1-\hat{\alpha}^2)$ is greater than about two (for large T, e.g, > 100), the sampler should be rerun with a larger value of I. Van Dyk and Jiao (2014) discuss other methods for choosing and validating I in the general MH within PCG sampler. (The choice of I is less important when implementing the fully Bayesian method, see Section 4.3.2.)

4.2.2 Optimizing the Iterated Pragmatic Bayesian Sampler

Given one certain A, the iterated sampler can jump across the parameter space, but then delivers M draws of θ that show relatively high correlation. We can characterize this behavior in the term of the *within-A* variance, which is $Var(\theta|Y, A)$ (i.e. the variance for a fixed A). To get new draw within one A, the computational time is relatively small. Once we resample a new A, a large constant computational time is needed to set up proposal distribution in `PyBLoCXS` algorithm. And the conditional mean, $E(\theta|A, Y)$, shifts to that of the newly sampled A, that is according to the *between-A* variance, which is the variability of $E(\theta|A, Y)$ as a function of A.

Thus, the variance of sample chain contains two components, within-A variance and between-A variance. The M correlated draws of θ in Step 3 vary according to within-A variance, and the conditional means, $E(\theta|A, Y)$, vary according to between-A variance.

We now turn to the question of choosing the optimal value of M. There is a tradeoff between choosing a small value of M to fully explore the posterior distribution and choosing a large

M to improve the computational efficiency. Put another way, there is a tradeoff to distribute total number of iterations between more outer iterations and more inner iterations. The fact that the computing cost for one inner iteration is much less than of the cost for one outer iteration complicates the tradeoff.

To balance this tradeoff, we aim to minimize $Var(\bar{\theta})$ with the fixed CPU time, where $\bar{\theta}$ is the sample mean of the chain,

$$\bar{\theta} = \frac{1}{TM} \sum_{t=1}^{T} \sum_{m=1}^{M} \theta^{(tM+m)} \tag{4.2}$$

Let $\bar{\theta}^t$ be the sample mean within one common $A^{(t)}$

$$\bar{\theta}^t = \frac{1}{M} \sum_{m=1}^{M} \theta^{(tM+m)} \tag{4.3}$$

From Equations (4.3) and (4.2), we have

$$\bar{\theta} = \frac{1}{T} \sum_{t=1}^{T} \bar{\theta}^t \tag{4.4}$$

Because of MH within PCG Sampler, $A^{(t)}$ is sampled from marginal distribution $\pi(A)$, and $\theta^{(tM+1)}, \ldots, \theta^{(tM+M)}$ are sampled only on $A^{(t)}$. Thus, $\bar{\theta}^t$ is independent to each other for different t.

$$Var(\bar{\theta}) = \frac{1}{T^2} \sum_{t=1}^{T} Var(\bar{\theta}^t) \tag{4.5}$$

Focus on one individual component $Var(\bar{\theta}^t)$ in Equation (4.5), we have

$$Var(\bar{\theta}^t) = E[Var(\bar{\theta}^t | A^{(t)})] + Var[E(\bar{\theta}^t | A^{(t)})] \tag{4.6}$$

We assume that $\theta^{(tM+1)}, \ldots, \theta^{(tM+M)}$ are AR(1) correlated. ρ_t is autocorrelation of the Monte Carlo chain for the standard method with $A = A^{(t)}$. When M is sufficiently large, incorporating correlation between $\theta^{(tM+1)}, \ldots, \theta^{(tM+M)}$, we have:

$$Var(\bar{\theta}^t|A^{(t)}) \approx \frac{\sigma^2_{\theta|A^{(t)}}}{M}\frac{1+\rho_t}{1-\rho_t} \tag{4.7}$$

We also assume $\rho_t \equiv \rho$ and $\sigma^2_{\theta|A^{(t)}} \equiv \sigma^2_{\theta|A} = W$, where W stands for within-A variance. This assumption essentially means that changing A only affects $E(\theta|A)$, but not $Var(\theta|A)$. In reality, this assumption holds in general. Thus,

$$E[Var(\bar{\theta}^t|A^{(t)})] \approx \frac{W}{M}\frac{1+\rho}{1-\rho} \tag{4.8}$$

Also, the other part of Equation (4.6) becomes:

$$Var[E(\bar{\theta}^t|A^{(t)})] = Var[E(\theta|A^{(t)})] = B \tag{4.9}$$

Here, B stands for between-A variance.

Place Equation (4.8) and (4.9) into Equation (4.5), we have

$$Var(\bar{\theta}) = \frac{1}{T}[B + W\frac{1+\rho}{M(1-\rho)}] \tag{4.10}$$

Let τ_1 be opportunity cost of the computational time once we have a new sampled A. It contains the time to set up new proposal distribution in `PyBLoCXS` and the time to sample the first discarded $I-1$ θ draws in Step 2. Let τ_2 be the computational time for each sub-iteration within Step 3.

In practice, since θ is a multivariate variable, we cannot simply minimize $Var(\bar{\theta})$. Usually, we pick the most important variable in θ, such as Γ in our example. With a fixed, total CPU time $\tau = T\tau_1 + TM\tau_2$, to minimize $Var(\bar{\Gamma})$, we have the analytical form of the optimal M.

$$M = \sqrt{\frac{\tau_1 W_\Gamma (1+\rho)}{\tau_2 B_\Gamma (1-\rho)}}, \tag{4.11}$$

Where W_Γ and B_Γ are the within-A variance and between-A variance for Γ. Note that the optimized M is not related to total CPU time τ.

Computing the optimal M using Equation (4.11) requires pre-estimates for T_1, T_2, ρ, W and B. In practice, we can firstly run T_0(about 20) affective area curves and M_0(about 20) sub-iterations for θ in order to get a rough estimate for T_1, T_2, ρ, W and B in order to use Equation (4.11).

The formulas for B and W can be found in Section 4.2.3. Here, we give the formula to estimate for ρ.

$$\hat{\rho} \approx \frac{1}{T_0} \sum_{t=1}^{T_0} \frac{\sum_{m=1}^{M_0-1} (\Gamma^{(tM_0+m)} - \bar{\Gamma}^t)(\Gamma^{(tM_0+m+1)} - \bar{\Gamma}^t)}{\sum_{m=1}^{M_0} (\Gamma^{(tM_0+m)} - \bar{\Gamma}^t)^2}$$

Back to the simulation example of Section 3.2, it can give us a general idea about how Iterated Pragmatic Bayesian works in reality. We can estimate $\hat{\tau}_1 = 6.2sec$, $\hat{\tau}_2 = 0.042sec$, $\hat{\rho} = 0.87$, $\hat{B}_\Gamma = 0.0021$, $\hat{W}_\Gamma = 0.000107$. Use the Equation (4.11), the optimal M is approximately equal to 10.

4.2.3 Computing Fitted Value and Error Bars with the Iterated Sampler

Once the iterations are obtained, multiple imputation combing rules are applied to compute the mean and error bars of θ. *Multiple Imputation* was first designed to handle missing data (Rubin (1987)). In multiple imputation, a number of Monte Carlo replications of the missing data is used to represent the the statistical qualities of the missing values. Here, we treat the effective areas curve as the missing data and use the multiple imputation combing rules to compute the within-A variance of θ and the between-A variance of θ. The mean estimate of θ is simply computed as the average of the samples:

$$\bar{\theta} = \frac{1}{TM} \sum_{t=1}^{T} \sum_{m=1}^{M} \theta^{(tM+m)} \tag{4.12}$$

The estimate of the within-A variance is calculated as:

$$\widehat{W} = \frac{1}{T(M-1)} \sum_{t=1}^{T} \sum_{m=1}^{M} (\theta^{(tM+m)} - \bar{\theta}^{t})^2 \tag{4.13}$$

The estimate of the between-A variance is calculated as:

$$\widehat{B} = \frac{1}{T-1} \sum_{t=1}^{T} (\bar{\theta}^{t} - \bar{\theta})(\bar{\theta}^{t} - \bar{\theta})^{\top} \tag{4.14}$$

The estimate of the total variance of θ is

$$\widehat{Var}(\theta) = \widehat{W} + (1 + \frac{1}{T})\widehat{B} \tag{4.15}$$

Here, the term $\frac{1}{T}$ accounts for the small number of imputations.

The iterated pragmatic Bayesian sampler is essentially one realization on application of multiple imputation. It uses the first two moments to obtain fitted values and error bars.

Since it relies on moment calculation, it may produce misleading results if the posterior distribution is highly non-Gaussian, e.g., it has multiple modes. The advantage of iterated pragmatic Bayesian sampler is its efficiency in practice. With constant CPU time, it provides the optimal number of samples to minimize $Var(\bar{\theta})$, which means the estimate of $\bar{\theta}$ from iterated pragmatic Bayesian sampler is more accurate and reliable.

Notice that the application of multiple imputation in iterated pragmatic Bayesian method is different from the Multiple Imputation method in Lee *et al.* (2011). The multiple imputation used by Lee *et al.* (2011), assumes that the parameter estimates $\hat{\theta}$ from sherpa fit is the mean of $p(\theta|Y, A)$. That's not true in general, since sherpa fit aims to find the mode of $p(\theta|Y, A)$. Lee *et al.* (2011) has not examined this assumption. Iterated pragmatic Bayesian method, on the other hand, uses only parameter estimates from sherpa fit for metropolis-hasting rules. Thus, it does not have this limitation.

4.3 Fully Bayesian Samplers

4.3.1 A Fully Bayesian MH within Gibbs Sampler

In order to construct an MCMC samplar with stationary distribution $p_{\text{fB}}(\theta, A|Y)$, we can use a two-step Gibbs sampler.

Fully Bayesian Gibbs Sampler :

For $t = 0, 1, 2, \ldots, T$,

Step 1: Sample $A^{(t+1)} \sim p(A|\theta^{(t)}, Y)$

Step 2: Sample $\theta^{(t+1)} \sim p(\theta|A^{(t+1)}, Y)$

Unlike the pragmatic Bayesian sampler, the fully Bayesian samples the effective area curve from its conditional posterior distribution, in Step 1. Although we cannot simulate directly from the conditional distribution, we can use a Mixed Metropolis and Independence Sampler. For Step 2, we can still directly apply `PyBLoCXS` to accomplish the sampling θ given A. We update θ same as in the pragmatic Bayesian samplers, except no inner iteration is required in Step 2 because this is a standard MH within Gibbs sampler. There is no partial collapse in Step 1, as it aims to simulated $p(A|\theta, Y)$ rather than $\pi(A)$.

The resulting sampler is:

MH within Fully Bayesian Gibbs Sampler :

For $t = 0, 1, ..., T$

Step 1: Sample $A^{(t+1)} \sim \mathcal{K}(A|A^{(t)}; Y, \theta^{(t)})$

Step 2: Sample $\theta^{(t+1)} \sim \mathcal{K}_{\text{pyB}}(\theta|\theta^{(t)}; Y, A^{(t+1)})$

MH within Gibbs Sampler is a very standard and easy algorithm in practice. If we consider these two steps as an entire unit, it essentially draws new samples $\{A^{(t+1)}, \theta^{(t+1)}\}$, from the multivariate kernel $\mathcal{K}(A, \theta|A^{(t)}, \theta^{(t)})$. Thus, it can also be regarded as an extended case of Metropolis-Hastings and preserves the nice convergence property as Metropolis-Hastings.

Now, we focus on the detail of the effective area curve update in Step 1 using a Mixed Metropolis and Independence Sampler.

Step 1A: Randomly sample u_1 from standard uniform distribution.

If $u_1 < p_m$, go to Step 1B, Metropolis Sampler

else, go to Step 1C, Independence Sampler

Step 1B: For $j = 1, ..., J$, Sample $e_j^{\text{prop}} \sim \mathcal{N}(\mu = e_j^{(t)}, sd = \sigma_e)$

set $e^{\text{prop}} = (e_1^{\text{prop}}, ..., e_J^{\text{prop}})$ and $A^{\text{prop}} = \bar{A} + (A_0^* - A_0) + \sum_{j=1}^{J} e_j^{\text{prop}} r_j v_j$.

$$\alpha = \frac{p(A^{\text{prop}}|Y, \theta^{(t)})}{p(A^{(t)}|Y, \theta^{(t)})}$$

go to Step 1D

Step 1C: For $j = 1, ..., J$, Sample $e_j^{\text{prop}} \sim \mathcal{N}(0, 1)$

set $e^{\text{prop}} = (e_1^{\text{prop}}, ..., e_J^{\text{prop}})$ and $A^{\text{prop}} = \bar{A} + (A_0^* - A_0) + \sum_{j=1}^{J} e_j^{\text{prop}} r_j v_j$.

$$\alpha = \frac{p(A^{\text{prop}}|Y, \theta^{(t)})Q(e^{(t)})}{p(A^{(t)}|Y, \theta^{(t)})Q(e^{\text{prop}})},$$

$Q(e^{\text{prop}})$ is the jumping rule, and here is simple standard normal density

go to Step 1D

Step 1D: Randomly sample u_2 from the standard uniform distribution.

$$(e^{(t+1)}, A^{(t+1)}) = \begin{cases} (e^{\text{prop}}, A^{\text{prop}}) & \text{if } u_2 < \alpha \\ (e^{(t)}, A^{(t)}) & \text{otherwise} \end{cases}.$$

In Step 1A, the tuning parameter p_m is the proportion of proposals that are using Metropolis Sampler. The Metropolis Sampler usually proposes the values near the previous draws, while the Independence Sampler allows to jump across the parameter space. Running the Independence sampler on its own, however can get stuck when A reaches high-density area where $p(A^{\text{prop}}|Y, \theta)/Q(e^{\text{prop}})$ is large. Therefore, a mixture of Metropolis Sampler and Metropolis-Hasting Sampler allows the samples to mix and results in better-performance chain. In Step 1B, we use normal distribution as Metropolis rules with $sd = \sigma_e$, where σ_e is a tuning parameter that determines the step size of Metropolis Sampler. This symmetric proposal is also known as random walk proposal. In Step 1C, we usually set $Q(e^{\text{prop}})$ as the prior distribution of e, that is the standard normal distribution. This proposal is an over-dispersed distribution, which is critical for independence sampler. Lastly, in order to evaluate the conditional posterior probability $p(A|Y,\theta)$, we can use the Bayesian approach:

$$p(A|Y,\theta) \propto L(Y|A,\theta)\pi(A),$$

Unfortunately, the fully Bayesian using MH within Gibbs sampler involves several tuning parameters, and the value of which can highly influence the performance of the resulting chain.(Note, there are two tuning parameters in Step 1, p_m and σ_e, and there are also three tuning parameters in `PyBLoCXS` of Step 2, see more in Appendix, A) Achieving good convergence requires significant trial and error in the setting of the tuning parameters. Moreover, the tuning parameters must be re-calibrated for every different data set, which limits the use of this sampler in practice. For instance, small data sets usually do not provide much information about the choice of effective area curve. In this case, picking smaller value of p_m is necessary in order to let the sampler rely more on the independence sampler.

In some cases it is nearly impossible to obtain samples adequately mixed with the MH within Gibbs sampler. In the next section, we describe a method that is much more efficient.

4.3.2 Using the Pragmatic Posterior as a Proposal Distribution

In this section, we propose the main algorithm for fully Bayesian inference. As we can see from Figure 3.1, the parameter estimates from the pragmatic Bayesian method have larger variance than those of the fully Bayesian method. This is because the pragmatic Bayesian method samples the effective area curve from the entire range of its prior distribution, while the fully Bayesian method aims to select those that are most consistent with the data. Thus, we expect the range of the samples of e and θ under the pragmatic Bayesian method to be a superset of that under the fully Bayesian method. With this in mind, we can construct an independence sampler for both the effective area curve and source parameter simultaneously, with the proposal distribution being the pragmatic Bayesian posterior distribution.

$$
\begin{aligned}
Q_{\mathrm{pB}}(\theta, A) = p_{\mathrm{pB}}(\theta, A|Y) &= p(\theta|A, Y)\pi(A) \\
&= \frac{L(Y|\theta, A)p(\theta|A)}{p(Y|A)}\pi(A) \qquad (4.16)
\end{aligned}
$$

The difficulty of using Equation (4.16) in independence sampler is that we can not evaluate the pragmatic Bayesian posterior. It involves the calculation of normalizing constant, $p(Y|A)$ in Equation (4.16). The Pragmatic Bayesian posterior is actually a doubly-intractable distribution. An interesting fact is that we can still use pragmatic Bayesian MCMC algorithm to sample from the pragmatic Bayesian posterior (Section 2.4). The typical problem is to find ways to sample doubly-intractable distributions. Our problem seems simpler, but is still intractable. More discussion about the doubly-intractable distributions and several solutions to sample doubly-intractable distributions can be found in Murray *et al.* (2012) and Liang (2010).

4.3.3 A Gaussian Approximation to the Pragmatic Posterior Distribution

Recognizing that the proposal distribution of the independence sampler can be any distribution as long as it can cover the fully Bayesian posterior $p_{\mathrm{fB}}(\theta, A|Y)$ nicely, we choose that the proposal distribution such that its mass is in the same location with $p_{\mathrm{fB}}(\theta, A|Y)$ and its tails are at least as heavy as $p_{\mathrm{fB}}(\theta, A|Y)$. Thus we suggest using a multivariate normal distribution $Q_{\mathrm{MVN}}(\theta, A)$ to fit a sample obtained from the pragmatic Bayesian posterior. The sample of the pragmatic Bayesian could be either from original pragmatic Bayesian sampler or from iterated pragmatic Bayesian Sampler. Both will work equally well as an over-dispersed approximation of $p_{\text{fB}}(\theta, A|Y)$. In practice, samples from iterated pragmatic Bayesian sampler is suggested because of its efficiency, which is discussed in Section 4.2. Suppose we have the pragmatic Bayesian samples, $\{(e_{\text{pB}}^{(1)}, \theta_{\text{pB}}^{(1)}), ..., (e_{\text{pB}}^{(T)}, \theta_{\text{pB}}^{(T)})\}$

We set the multivariate normal distribution $Q_{\text{MVN}}(\theta, e)$ to be,

$$\begin{pmatrix} \theta \\ e \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \hat{\mu}_\theta \\ \mu_e \end{pmatrix}, \begin{pmatrix} \hat{\Sigma}_{11} & \hat{\Sigma}_{12} \\ \hat{\Sigma}_{21} & \Sigma_{22} \end{pmatrix}\right), \tag{4.17}$$

Since e is sampled from the standard normal distribution in the pragmatic Bayesian sampler (Recall that under the pragmatic Bayesian model, e is simulated according to its prior distribution), $\mu_e = 0$ and $\Sigma_{22} = I$. We set $\hat{\mu}_\theta$ and $\hat{\Sigma}_{11}$ as the sample mean and sample variance of θ obtained from the pragmatic Bayesian sample and set $\hat{\Sigma}_{12}$ as the sample covariance between θ and e.

$$\begin{aligned} \hat{\mu}_\theta &= \frac{\sum_{t=1}^{T} \theta_{\text{pB}}^{(t)}}{T} \\ \hat{\Sigma}_{11} &= \frac{\sum_{t=1}^{T} (\theta_{\text{pB}}^{(t)} - \hat{\mu}_\theta)(\theta_{\text{pB}}^{(t)} - \hat{\mu}_\theta)^\top}{T} \\ \hat{\Sigma}_{12} &= \frac{\sum_{t=1}^{T} e_{\text{pB}}^{(t)}(\theta_{\text{pB}}^{(t)} - \hat{\mu}_\theta)^\top}{T} \\ \hat{\Sigma}_{21} &= \hat{\Sigma}_{12}^\top \end{aligned} \tag{4.18}$$

The conditional distribution of θ given e, written as $Q_{\text{MVN}}(\theta|e)$, is,

$$Q_{\text{MVN}}(\theta|e) \sim \mathcal{N}(\hat{\mu}_\theta + \hat{\Sigma}_{12}e, \hat{\Sigma}_{11} - \hat{\Sigma}_{12}\hat{\Sigma}_{21}) \tag{4.19}$$

Thus, sampling from proposal distribution $Q_{\text{MVN}}(\theta, e)$ is straightforward. First, we sample

e from the standard multivariate normal distribution. Next we sample θ from $Q_{\mathrm{MVN}}(\theta|e)$. Both steps only involve sampling from multivariate normal distribution.

Notice that using $Q_{\mathrm{MVN}}(\theta, A)$ as the proposal distribution leads to an Independence Sampler, as the current iteration does not depend on the previous iteration. A slight modification can be made. We can still use a random walk sampler for e, as is done in Step 1B of Fully Bayesian algorithm using Gibbs sampling. The new proposal distribution can be written:

$$\begin{aligned} Q'_{\mathrm{MVN}}(\theta, e|\theta^{(t)}, e^{(t)}) &= Q'_{\mathrm{MVN}}(\theta, e|e^{(t)}) \\ &= Q_{\mathrm{MVN}}(\theta|e)Q'_{\mathrm{MVN}}(e|e^{(t)}) \qquad (4.20) \end{aligned}$$

$Q'_{\mathrm{MVN}}(e|e^{(t)})$ can simply be set to a normal distribution with mean equal to $e^{(t)}$. Overall, Q' is not a random walk sampler, it involves $Q_{\mathrm{MVN}}(\theta|A)$. We call it a Partial Random Walk Sampler.

4.3.4 A Fully Bayesian Sampler with A Pragmatic Proposal

Combing these two proposal distributions, we can conduct a mixture of Independence Sampler and Partial Random Walk Sampler for the fully Bayesian method.

Pragmatic Proposal Sampler:

For $t = 1, 2, ..., T$

Step 1: Randomly sample u_1 from a standard uniform distribution.

If $u_1 < p_m$, go to Step 2, Partial Random Walk Sampler else, go to Step 3, Independence Sampler

Step 2: For $j = 1, ..., J$, Sample $e_j^{\text{prop}} \sim \mathcal{N}(\mu = e_j^{(t)}, sd = \sigma_e)$

Set $e^{\text{prop}} = (e_1^{\text{prop}}, ..., e_J^{\text{prop}})$ and $A^{\text{prop}} = \bar{A} + (A_0^* - A_0) + \sum_{j=1}^{J} e_j^{\text{prop}} r_j v_j$.

Sample $\theta^{\text{prop}} \sim Q_{\text{MVN}}(\theta|e^{\text{prop}})$

$$\alpha = \frac{p(A^{\text{prop}}, \theta^{\text{prop}}|Y) Q_{\text{MVN}}(\theta^{(t)}|e^{(t)})}{p(A^{(t)}, \theta^{(t)}|Y) Q_{\text{MVN}}(\theta^{\text{prop}}|e^{\text{prop}})},$$

go to Step 4

Step 3: For $j = 1, ..., J$, Sample $e_j^{\text{prop}} \sim \mathcal{N}(0, 1)$

Set $e^{\text{prop}} = (e_1^{\text{prop}}, ..., e_J^{\text{prop}})$ and $A^{\text{prop}} = \bar{A} + (A_0^* - A_0) + \sum_{j=1}^{J} e_j^{\text{prop}} r_j v_j$.

Sample $\theta^{\text{prop}} \sim Q_{\text{MVN}}(\theta|e^{\text{prop}})$

$$\alpha = \frac{p(A^{\text{prop}}, \theta^{\text{prop}}|Y) Q_{\text{MVN}}(\theta^{(t)}, e^{(t)})}{p(A(t), \theta^{(t)}|Y) Q_{\text{MVN}}(\theta^{\text{prop}}, e^{\text{prop}})},$$

go to Step 4

Step 4: Randomly sample u_2 from a standard uniform distribution.

$$(e^{(t+1)}, A^{(t+1)}, \theta^{(t+1)}) = \begin{cases} (e^{\text{prop}}, A^{\text{prop}}, \theta^{\text{prop}}) & \text{if } u_2 < \alpha \\ (e^{(t)}, A^{(t)}, \theta^{(t)}) & \text{otherwise} \end{cases}.$$

By taking the information of the pragmatic Bayesian sampler into the proposal distribution, this algorithm has relative higher acceptance rate than the MH within Gibbs fully Bayesian algorithm. Moreover, this algorithm no longer needs to use the sampler in `PyBLoCXS` which makes the algorithm much more efficient. Only two tuning parameters are needed to sample A (p_m and σ_e). In practice, the default setting, $p_m = 0.5$ and $\sigma_e = 0.1$ works well for most data sets; this is because we have already picked much more efficient proposal distribution afterwards in simulating θ, by using $Q_{\text{MVN}}(\theta|e)$. In contrast, MH within Gibbs fully Bayesian

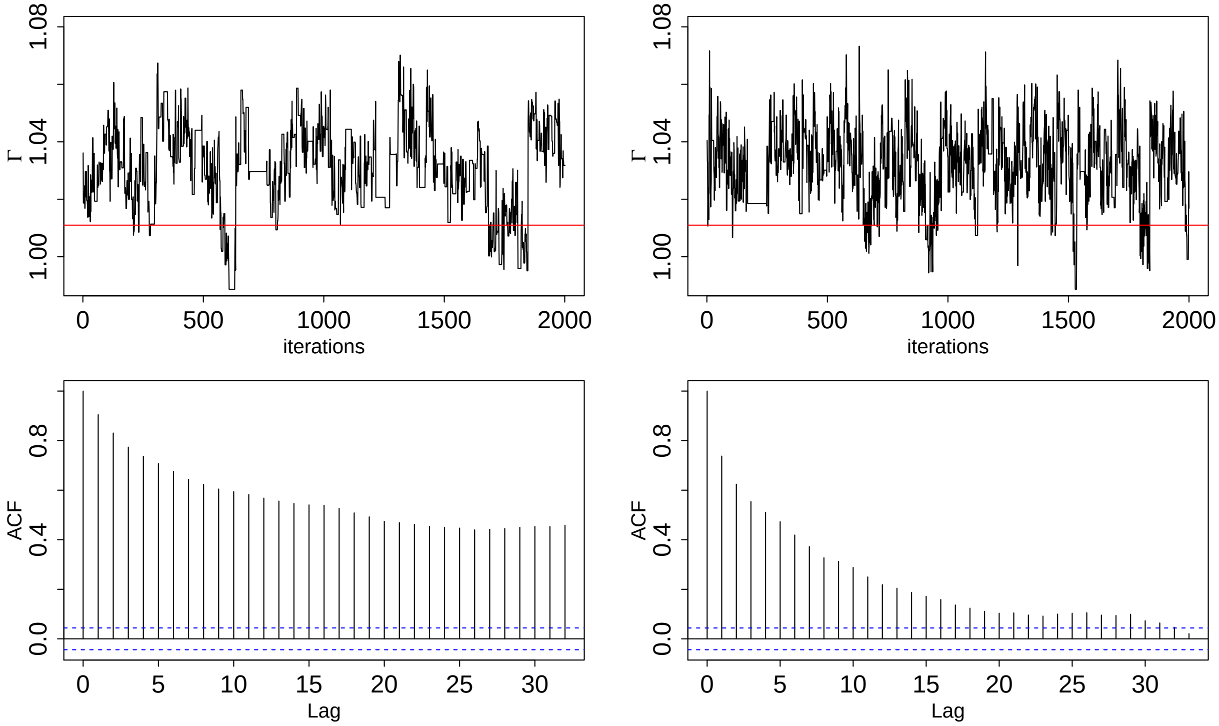

Figure 4.2: Comparison of two fully Bayesian MCMC chains. Two fully bayesian samplers are conducted for the simulated data set described in Section 3.2. The left column is the MCMC draws from fully Bayesian using Gibbs sampler. The right column is the MCMC draws from the pragmatic proposal sampler. The bottom row are the autocorrelation function plots for the two samplers. The the pragmatic proposal sampler exhibits better convergence. As the left bottom panel shows, the draws is almost uncorrelated when $Lag = 30$.

algorithm needs five tuning parameters to set up.

Compared to MH within Gibbs fully Bayesian sampler Section 4.3.1, the Pragmatic Proposal Sampler has a dominant advantage in the performance of MCMC chain. Here, we compare these two samplers by fitting the simulated data set, that is described in Section 3.2. For the MH within Gibbs sampler, we set $p_m = 0.5$ and $\sigma_e = 0.1$ for the Step 1, and $scale^{pyb} = 0.5, df^{pyb} = 4, p_m^{pyb} = 0.5$ for the `PyBLoCXS` of Step 2. For the pragmatic proposal sampler, we first need to draw samples from iterated pragmatic Bayesian sampler, in which we use the optimal tuning parameters $M = 10$ (see in Section 4.2.2). We then use pragmatic proposal sampler with $p_m = 0.5$ and $\sigma_e = 0.1$.

We compare these two fully Bayesian samplers using two criteria–the computational time and the convergence rate. For the MH within Gibbs sampler, because of the existence of `PyBLoCXS` in Step 2, 2000 iterations cost about 2-hour CPU time. For the pragmatic proposal sampler, 2000 iterations for iterated pragmatic Bayesian sampler and the pragmatic proposal sampler costs about 30-minute and 10-minute CPU time, respectively. Therefore, the pragmatic proposal sampler is more efficient than the MH within Gibbs sampler.

Figure 4.2 shows the performance of two chains from these two samplers. The left column is the MCMC draws from MH within Gibbs sampler, and the right column is the MCMC draws from the pragmatic proposal sampler. It shows that the draws form MH within Gibbs sampler performs more "sticky", and thus leads into a slower convergence rate. The two bottom panels are the autocorrelation function plots for the two samplers. The autocorrelation of pragmatic proposal sampler reduces much faster than that of the Gibbs sampler. When $Lag = 30$, the draws is almost uncorrelated.

Overall, the pragmatic proposal sampler not only saves more than half the computational time, but also improves the chain convergence rate substantially. Besides, it does not need to apply the `PyBLoCXS` sampler, and thus avoids the problem of tuning parameters. Therefore, in practice, the combination of iterated pragmatic Bayesian sampler and the pragmatic proposal sampler for fully Bayesian is recommended.

In Chapter 5, we will also compare two versions of Pragmatic Proposal Sampler, one based on MH within PCG Sampler of Lee *et al.* (2011) of Section 4.1 and one based on iterated pragmatic Bayesian sampler of Section 4.2.1 in the simulation study. For the real data analysis, we will use the combination of iterated pragmatic Bayesian sampler and the pragmatic proposal sampler for fully Bayesian as the default method.

Chapter 5

Numerical Performance

In this chapter, we will first use a series of simulation studies to identify the circumstances under which the fully Bayesian method is advantageous. Five methods–Standard method, pragmatic Bayesian method, iterated pragmatic Bayesian method, fully Bayesian method based on pragmatic Bayesian, and fully Bayesian method based on iterated pragmatic Bayesian–will be compared. Then, frequency analysis will be conducted to provide more accurate results. Last, we apply the standard method, iterated pragmatic Bayesian method and fully Bayesian method based on samples from iterated pragmatic Bayesian method to real data to demonstrate the advantages of the fully Bayesian method. We consider power-law and multi-thermal sources observed with ACIS-S, and a nominal blackbody source observed with Chandra HRC-S/LETG (HRC stands for High Resolution Camera, and LETG stands for High Energy Transmission Grating.).

Table 5.1: The eight simulations used to compare the standard, pragmatic Bayesian, and fully Bayesian methods.

	Effective Area		Nominal Counts		Spectal Model	
	Default*	Extreme**	10^5	10^4	Hard†	Soft‡
Simulation I		X	X		X	
Simulation II		X	X			X
Simulation III		X		X	X	
Simulation IV		X		X		X
Simulation V	X		X		X	
Simulation VI	X		X			X
Simulation VII	X			X	X	
Simulation VIII	X			X		X

*The default effective area curve, A_0, from the calibration library of Drake *et al.* (2006)

**An extreme effective area curve , A_{ext}, from the calibration library of Drake *et al.* (2006)

†An absorbed powerlaw with $\Gamma = 2$, $N_{\text{H}} = 10^{23}/\text{cm}^2$

‡An absorbed powerlaw with $\Gamma = 1$, $N_{\text{H}} = 10^{21}/\text{cm}^2$

5.1 Simulation Study

We begin by replicating the same eight simulation studies of Lee *et al.* (2011), but this time using the fully Bayesian method. The data sets were simulated from an absorbed power-law source with three parameters (power-law index Γ, absorption column density N_H, and normalization) using the **fake_pha** routine in Sherpa. The first four data sets were all simulated without background contamination using the **XSPEC** model **wabs*powerlaw**, extremal effective area curve A_{ext} from the calibration sample of Drake et al. (2006), and a default RMF for ACIS-S. The power law parameter (Γ), column density (N_H), and the nominal counts are:

Simulation I: $\Gamma = 2$, $N_H = 10^{23} cm^{-2}$, and 10^5 counts;

Simulation II: $\Gamma = 1$, $N_H = 10^{21} cm^{-2}$, and 10^5 counts;

Simulation III: $\Gamma = 2$, $N_H = 10^{23} cm^{-2}$, and 10^4 counts;

Simulation IV: $\Gamma = 1$, $N_H = 10^{21} cm^{-2}$, and 10^4 counts

The other 4 simulated data sets, Simulation V-VIII, have the exact same parameter setting, except that they are simulated from nominal default effective area curve, A_0. The simulation represents $2 \times 2 \times 2$ designs with three factors being (1) data simulated with an extremal effective area curve A_{ext} from $\mathcal{A}$ and with A_0 , (2) 10^5 and 10^4 photon counts, and (3) hard spectral powerlaw model with $\Gamma = 2, N_H = 10^{23}cm^{-2}$ and soft spectral powerlaw model with $\Gamma = 1, N_H = 10^{21}cm^{-2}$. The setting is illustrated in detail in Table 5.1. We comprehensively investigate the performance of the fully Bayesian method with regard to these three factors, and make the model comparison with the standard method and the pragmatic Bayesian method. These eight data sets are all fit with the default effective area curve, A_0. This means that first four data sets are fit with the wrong effective area curve, while the other four are using the right effective area curve. This enables us to investigate the effect of misspecification of A_0, where the degree of misspecification is consistent with the variability of the calibration library. Posterior distributions, intervals and fitted values for Γ computed with each of three methods run on each simulation appear in Figures 5.1 and 5.2. The standard, pragmatic Bayes and Fully Bayesian methods are represented by black, blue, and red curves and intervals, respectively. The posterior means (fitted values) are represented by a "×" and their 68.2% error bars are represented by horizontal bars. The joint posterior distribution of Γ and N_{H} under Simulation II for each of the three methods was discussed in Section 3.2 and appeared in Figure 3.1. In all cases, the pragmatic Bayesian method and iterated Pragmatic bayesian have almost identical results, same with the fully Bayesian method based on the pragmatic Bayesian and the fully Bayesian method based on iterated Pragmatic Bayesian. Thus, in Figures 5.1 and 5.2, we only plot the posterior density curves for the pragmatic Bayesian and the fully Bayesian method based on the pragmatic Bayesian.

In all eight simulations, the standard method exhibits significantly narrower error bars than the other methods, especially in the large count Simulations I, II, V, and VI. For the Simulation I–IV, however, its intervals miss the true value of Γ by a large margin. The pragmatic Bayesian method, by contrast, exhibits similar fitted values but much wider error bars, which reflects the variability in the fits resulting from different choices of A in the range of $\mathcal{A}$. Finally the fully Bayesian methods uses the data to exclude some A in the range of $\mathcal{A}$ that are inconsistent with the observed spectra. The result is optimal in that the fitted values shift toward the true value of Γ and the width of the error bars is narrow relative to those produced with the pragmatic Bayesian method. Thus, like the pragmatic Bayesian method, the fully Bayesian methods provides error bars that contain the true value at nearly the correct statistical rate (68.2% here), but these error bars are narrower than those provided by the pragmatic Bayesian method—especially when the counts are large. We conduct a large-scale simulation study in Section 5.2 to better quantify these trends.

For the Simulation VI-VIII, since we assume that the true effective area is known, as expected, the standard method performs the best. However, this assumption is unrealistic, as we never know the true effective area in reality.

The results of the simulations can be better understood by considering the statistical errors (due to Poisson fluctuations in the counts) and the systematic errors (due to misspecification of the effective area curve). The standard method ignores the latter sources of error and thus, not surprisingly, it provides narrower errors but can exhibit significant bias if the misspecification of A is substantial. Because it only considers statistical errors, the standard methods produces much narrower error bars for large-count spectra. The pragmatic Bayesian method, on the other hand, incorporates both statistical and systematic errors, resulting in significantly wider error bars. Because systematic errors do not dissipate as the sample size grows, the error bars produced by the pragmatic Bayesian method are not particularly sensitive to the number of counts. Lee *et al.* (2011) speculated that as the photon counts grow and the statistical errors become negligible, the error bars produced by the pragmatic methods will be entirely due to calibration uncertainty.

The power of the fully Bayesian method is that it actually uses the data to measure the systematic error. Put another way, it transforms systematic error into statistical error.

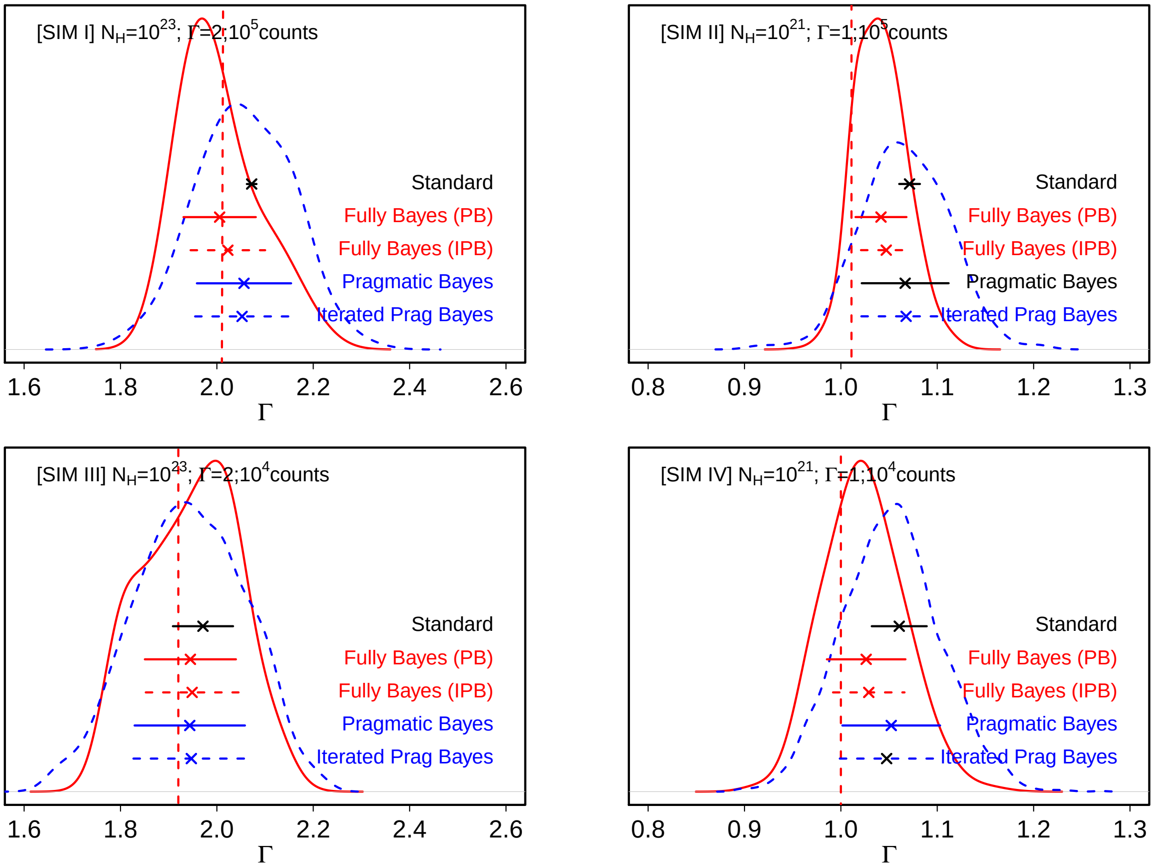

Figure 5.1: Results for Simulations I–IV. The panels show the posterior distributions (curves), fitted values (×), and 68.2% error bars (horizontal bars) for Γ. Results for the standard, pragmatic Bayesian, and fully Bayesian methods are plotted in black, blue, and red, respectively. (The plotted pragmatic Bayesian posterior distributions correspond to the MH within PCG Sampler of Lee *et al.* (2011) and the plotted fully Bayesian posterior distributions corresponds the fully Bayesian based on the MH within PCG Sampler.) The true value of Γ is given by the red broken vertical lines. These simulations consider the situation in which the default effective area curve is misspecified to a degree that is consistent with the variability of the the calibration library. Because the standard method uses this misspecified curve, it preforms poorly. Both the pragmatic and the fully Bayesian methods avoid assuming that A is known without error allowing them to preform better. Of these two, the fully Bayesian method provides both estimates of Γ that are closer to its true values and narrower error bars..

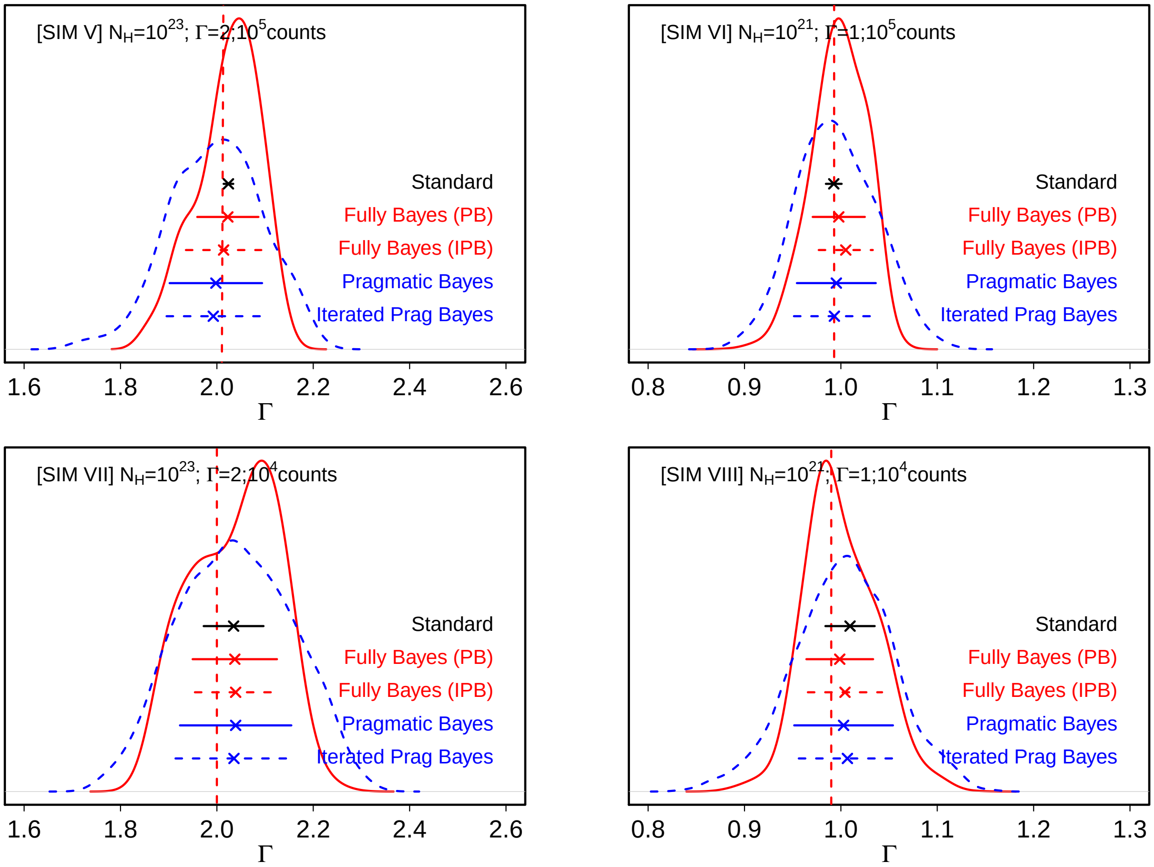

Figure 5.2: Results for Simulations VI–VIII. Since we assume that the true effective area is known in these four simulations, as expected, the standard method performs the best.

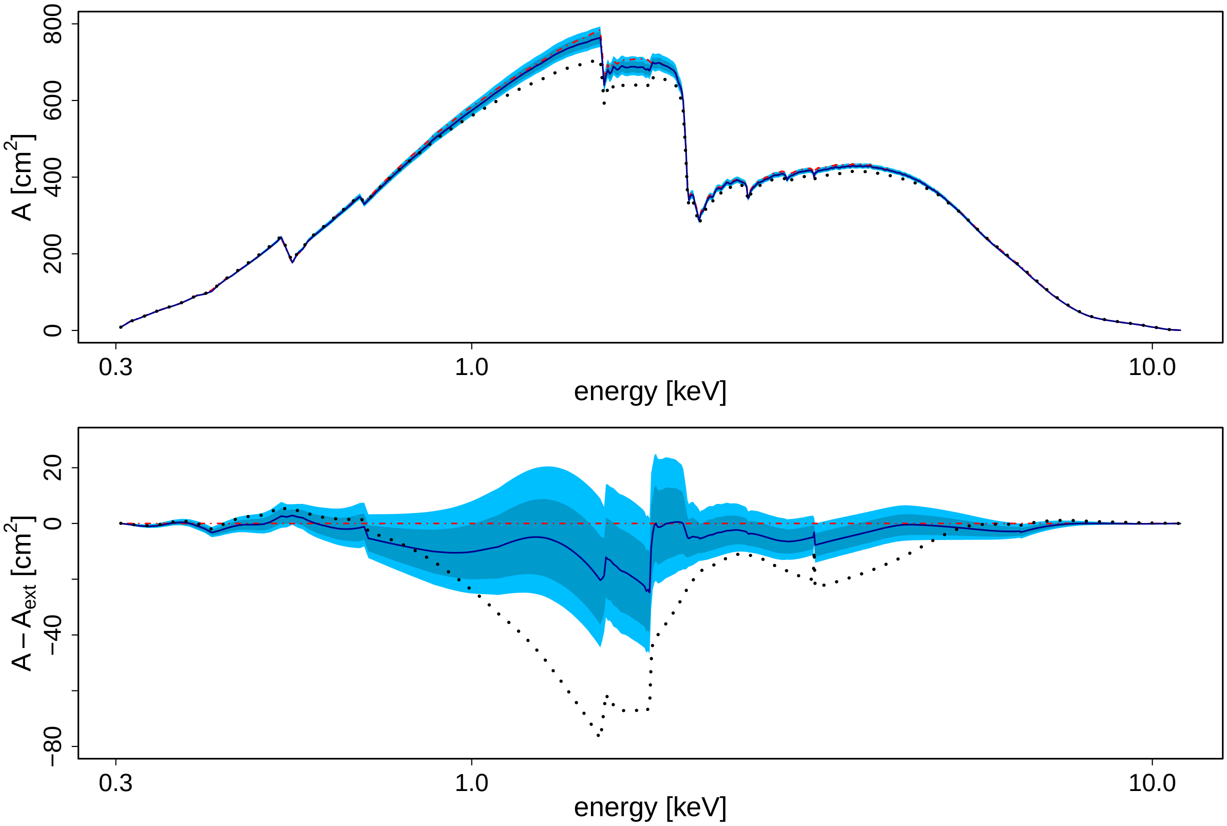

Figure 5.3: Fitting the effective area curve. The plots summarize $p_{\mathrm{fB}}(A \mid Y)$ for Simulation II. The true and the default effective area curves are plotted in red and black, respectively. The pointwise posterior distribution of A is plotted in blue, where the dark (light) blue area corresponds to the central 68.3% (90%) region of the posterior distribution; the pointwise posterior mean in plotted as a blue curve. The top panel is on the effective area scale and the bottom panel subtracts off the true effective area curve, A_{ext}, to magnify the differences. The true effective area curve is contained largely (almost entirely) in the light (dark) blue region. The posterior means (blue curve) is shifted from the prior mean to the true effective area curve (in red). Because we do not use the counts in bins below 0.3keV, we do not compute $p_{\mathrm{fB}}(A|Y)$ for these bins.

Thus, its error bars are wider than those provided by the standard method because they incorporate both sources of error, but their width descreases as the number of counts grows because the data are able to narrow the calibration uncertainty. This in turn allows us to go one step further and actually estimate A using its posterior distribution, $p_{\text{fB}}(A \mid Y)$.

We illustrate this in Figure 5.3 which shows how $p_{\text{fB}}(A \mid Y)$ differs from A_0 when fitting Simulation II. The shaded light blue region corresponds to a 90% pointwise posterior region for A and contains the true effective area curve, A_{ext} (plotted in red). The posterior region, however, is shifted toward A_0 (plotted in black), which serves as the prior mean for A. Thus, the prior distribution on A has a clear influence on its posterior distribution. A similar pattern can be seen in Figure 5.1. The fitted values of Γ are pulled from the true value toward the best value assuming $A = A_0$ as computed with the standard method. We emphasize, however, that the prior distribution on A used by the fully Bayesian method is in fact much *weaker* than what is commonly used in practice: the assumption that A is exactly equal to A_0. The standard method makes this assumption and as was demonstrated in Section 5.2, its fitted values exhibit significant bias when A_0 is misspecified.

Note that the components of e given Y in the fully Bayesian method are not independent of each other anymore, which makes the inference of A a bit difficult. We still have the property $\bar{A} = A(\bar{e}_1, ..., \bar{e}_J)$, which is a linear transformation of $A(e)$. But we cannot simply compute the quantile of A, as $A_q = A(e_{1,q}, ..., e_{J,q})$, where $e_{j,q}$ represents the qth quantile of e_j. The way we use for Figure 5.3 is making inference of the effective area one energy bin by one energy bin, using the samples of $A(e)$, rather than directly using e.

5.2 Frequency Evaluation

The $2 \times 2 \times 2$ simulation study described in Section 5.1 generated only one data set for each of the eight simulation studies. Part of the reason why Lee *et al.* (2011) limited the simulation in this way is because their MH within PCG sampler requires significant computational time. The iterated MH within PCG sampler that we propose does the same job with a fraction 1/9 of the computational costs. In this section we generate fifty spectra in each of the eight simulation settings described in Table 5.1 and fit each of the resulting 50×8 simulated spectra with the standard, pragmatic Bayesian, iterated pragmatic Bayesian, fully Bayesian based on pragmatic Bayesian and fully Bayesian based on iterated pragmatic Bayesian. The fitted values for Γ and their 68.2% error bars computed using the first ten spectra generated in each simulation appear in Figure 5.4 and 5.5. Numerical summaries of the entire simulation study appear in Table 5.2. The fully frequency analysis confirms the trends we observed in Section 5.1:

- The pragmatic Bayesian and the iterated pragmatic Bayesian perform almost the same. In practice, the iterated pragmatic Bayesian can actually replace the pragmatic Bayesian, since the CPU time of the iterated pragmatic Bayesian is about 1/9 time of the pragmatic Bayesian.

- No matter whether the fully Bayesian is based on pragmatic Bayesian or the iterated pragmatic Bayesian, the results of the fully Bayesian are almost the same.

- In all eight simulations, the standard method on average exhibits the narrowest error bars (mean error bars), the pragmatic Bayesian method has the widest, and those of the fully Bayesian method are in between.

- When A is correctly specified all three methods preform well, but the standard method is optimal, see Simulation V–VIII.

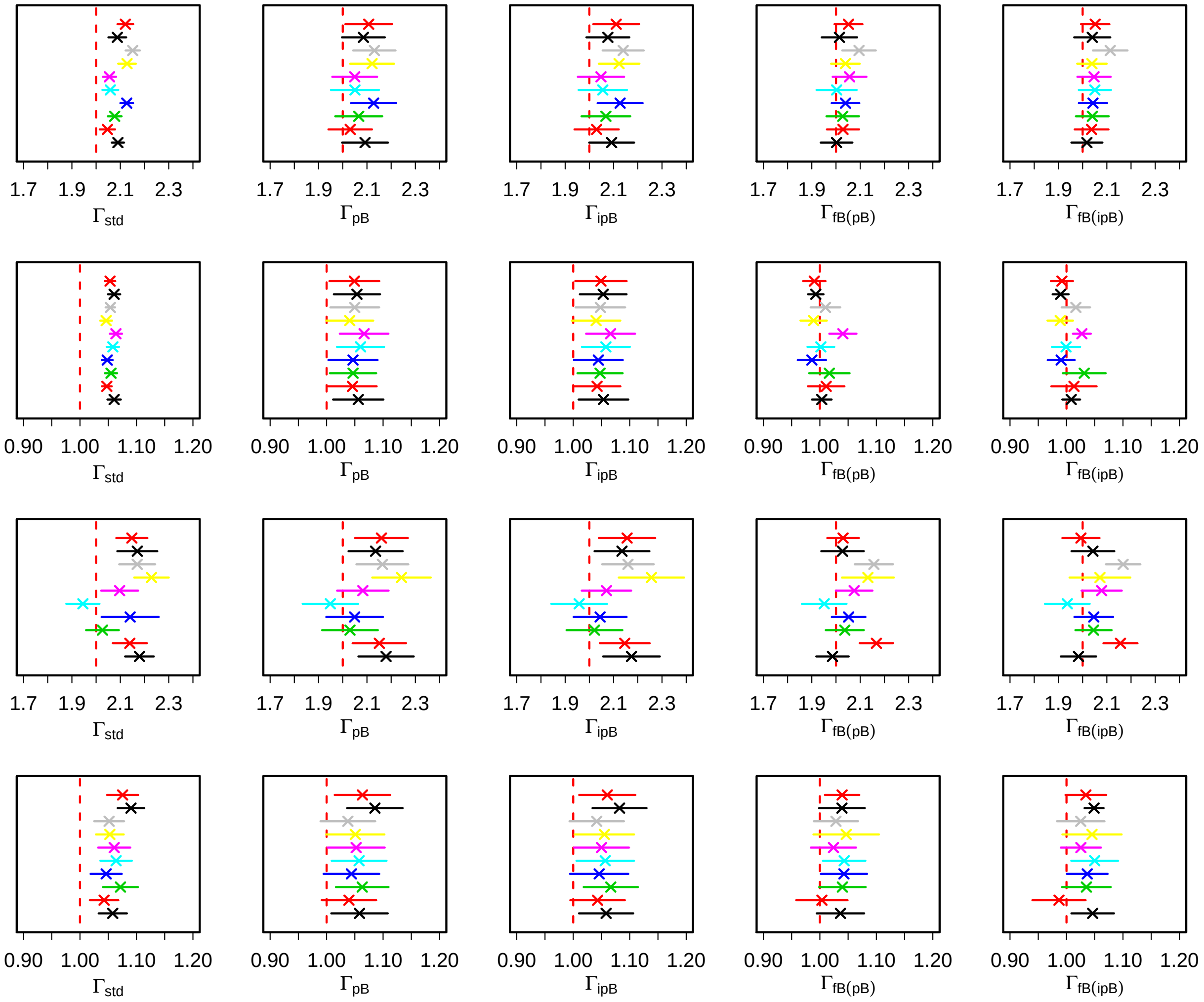

Figure 5.4: Frequency Analysis for Simulations I–IV. Each panel gives the fitted values and one σ error bars for Γ resulting from ten replicate simulations. The red broken vertical line in each panel indicated the true value of Γ. The four rows correspond to the four simulation settings and the five columns correspond to the standard, pragmatic Bayesian, iterated pragmatic Bayesian, fully Bayesian based on pragmatic Bayesian and fully Bayesian based on iterated pragmatic Bayesian, respectively. Owing to the misspecification of A_0, the standard and pragmatic Bayesian methods both exhibit significant bias. The pragmatic Bayesian methods adjusts for this bias with wider error bars, while the fully Bayesian method reduces the bias. Overall the fully Bayesian method is able to cover the true value of Γ more often with narrow errors bars than either of the other methods, especially with large-count spectra (i.e., Simulations I and II).

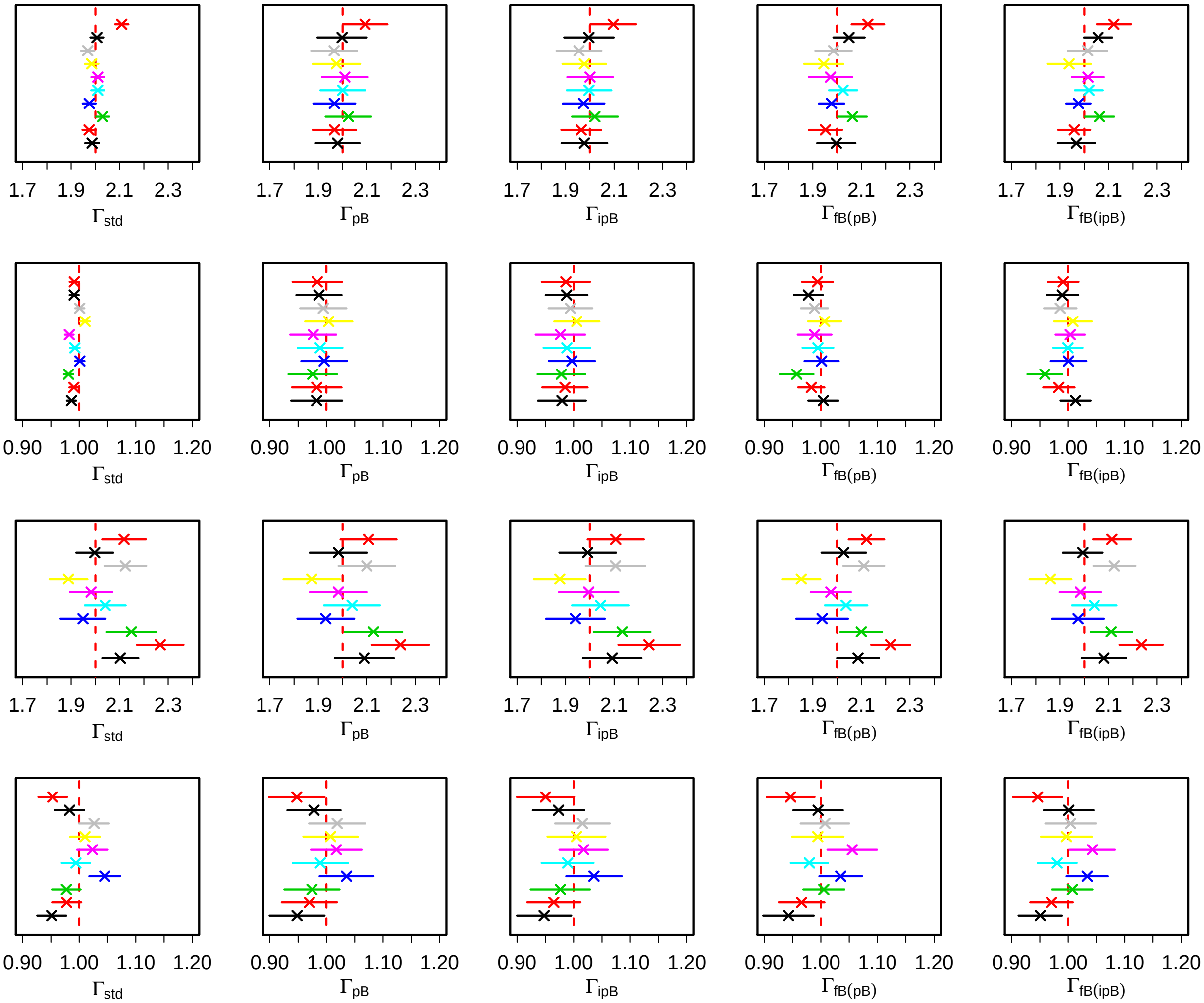

Figure 5.5: Frequency Analysis for Simulations V–VIII. These four simulation settings are designed to investigate the performance of three methods when fit with the right effective area curve. As expected, the standard method performs the best.

Table 5.2: Numerical results of frequency analysis.

	Standard					Pragmatic Bayesian					Fully Bayesian				
	cover-age[1]	mean error bars[2]	bias[3]	std error[4]	root mse[5]	cover-age	mean error bars	bias	std error	root mse	cover-age	mean error bars	bias	std error	root mse
Sim I[†]	0.00	0.029	0.092	0.029	0.097	0.62	0.093	0.082	0.030	0.088	0.76	0.065	0.035	0.040	0.052
Sim II[†]	0.00	0.010	0.056	0.007	0.056	0.80	0.042	0.053	0.008	0.054	0.78	0.024	0.002	0.017	0.016
Sim III	0.40	0.075	0.099	0.078	0.125	0.68	0.112	0.083	0.079	0.113	0.66	0.084	0.068	0.078	0.103
Sim IV	0.12	0.026	0.057	0.027	0.063	0.48	0.049	0.053	0.028	0.059	0.56	0.040	0.040	0.036	0.053
Sim V	0.70	0.028	-0.001	0.029	0.029	0.98	0.094	-0.012	0.029	0.031	0.92	0.066	-0.003	0.038	0.038
Sim VI	0.64	0.008	-0.009	0.008	0.008	1.00	0.040	-0.006	0.009	0.011	0.80	0.027	-0.009	0.010	0.014
Sim VII	0.66	0.084	0.021	0.099	0.102	0.78	0.116	0.018	0.100	0.101	0.66	0.86	0.018	0.102	0.102
Sim VIII	0.64	0.031	-0.000	0.032	0.031	0.84	0.049	-0.005	0.031	0.031	0.72	0.040	-0.004	0.034	0.034

[1] The proportion of 67% (one σ) intervals that contain the true value of Γ.

[2] The mean length of the one σ error bars.

[3] The mean of the fitted values of Γ minus its true value.

[4] The standard deviation of the fitted values of Γ.

[5] The square root of the mean of the squared deviations between the fitted value and the true value of Γ.

[†] Simulations I and II are highlighted in red because they show the largest gain in performance for the fully Bayesian method. Specifically, when the effective area curve is misspecified with a high-count spectra.

- In the more realistic situation in which there is uncertainty in A (i.e., Simulations I–IV) both the standard and pragmatic Bayesian methods exhibit substantial bias. The fully Bayesian method reduces this bias, resulting in the overall smallest mean square error.
- The advantage of the fully Bayesian method is most striking when A is misspecified and the data set is large (Simulations I and II). In this case the estimates produced by the fully Bayesian method have much smaller bias and root mean square error than those of the other two methods. Fully Bayesian intervals are much more likely to include the true value of Γ (coverage) than intervals based on a fixed effective area fit and the fully Bayesian error bars can be much narrower than the pragmatic Bayesian error bars. These effects dissipate with smaller data sets because substantial large data is required to narrow the calibration uncertainty.

5.3 Application to a Sample of Radio-Loud Quasars

We use a small sample of radio loud quasars to illustrate the relative performance of the standard, pragmatic Bayesian, and fully Bayesian methods. In radio loud quasars, X-ray emission originates in close vicinity to a supermassive black hole and is believed to be caused by either an accretion disk or a relativistic jet. This emission can be modeled with a Compton scattering process and the X-ray spectrum described using an absorbed power law:

$$\Lambda(E;\theta) = N\ E^{-\Gamma}\ e^{-\sigma(E)\,N_{\rm H}}\ \text{ photons cm}^{-2}\text{ sec}^{-1}\text{ keV}^{-1}\,, \tag{5.1}$$

where $\sigma(E)$ is the absorption cross-section, and the three model parameters are $\theta = (N, \Gamma, N_{\rm H})$ with N the normalization at 1 keV, Γ the photon index of the power law, and $N_{\rm H}$ the absorption column. X-ray spectra of the quasars were observed with the *Chandra* X-ray Observatory in 2002 (Siemiginowska *et al.*, 2008; Lee *et al.*, 2011). Standard data processing including source extraction and calibration was preformed using the `CIAO` software (*Chandra* Interactive Analysis of Observations).

The number of counts in the observed spectra vary between 8 and 5500. Following Lee *et al.* (2011), we excluded two of the spectra (ObsID 3099 which had 8 counts, and ObsID 836 which is better described by a thermal spectrum) and reanalyzed the remaining 15 using the standard, pragmatic Bayesian, and fully Bayesian methods. We account for background contamination using a background spectrum extracted over large annuli surrounding each source and a highly-structured background model that was originally fit to the blank-sky data provided by the *Chandra* X-ray Center, see Lee *et al.* (2011) for details. Only the normalization of the background model was fit in the individual spectral analyses. This approach was used for all but the two lowest-count spectra (< 45, both with short 5 ksec exposures), for which background was ignored. The sample of quasars was originally analyzed by Siemiginowska *et al.* (2008) who did not account for calibration uncertainty. A followup analysis accounted for calibration uncertainty using the pragmatic Bayesian method and resulted in substantially larger error bars for the high-count datasets (Lee *et al.*, 2011). As illustrated in Section 5.1, systematic errors due to calibration uncertainty swamp statistical errors for large data sets. For small data sets, however, the statistical errors may be much larger and the relative effect of calibration uncertainty is therefore less important. Here we reanalyze the same spectra with the fully Bayesian method and illustrate how it is able to deliver low-bias parameter estimates with smaller error bars than the pragmatic Bayesian method.

We fit each spectrum in three ways:

1. with the standard method,
2. with the pragmatic Bayesian method using the Iterated MH within PCG Sampler, and
3. with the fully Bayesian method using the Mixed Pragmatic Proposal Sampler.

With each of the three methods, we use the fifteen observation-specific default effective area curves, A_0^* corresponding to each spectra. For the two Bayesian methods, we use $J = 8$ in the PCA summary of the calibration library along with the A_0^* in Equation 2.4. When running the Iterated MH within PCG Sampler, we set $I = 10$ and $M = 10$; that is at each iteration we run `PyBLoCXS` $M + I - 1$ times, discarding the output of the first $I - 1$ runs and keeping the output of the final M runs. In the Mixed Pragmatic Proposal Sampler, the two proposal rules were used in equal proportion and with the random-walk proposal we set $\sigma = 0.1$.

Results are illustrated in Figures 5.6 and 5.7. Error bars for Γ computed under the pragmatic Bayesian ($\hat{\sigma}_{\text{pB}}$) and fully Bayesian ($\hat{\sigma}_{\text{fB}}$) methods are compared with those computed under the standard method ($\hat{\sigma}_{\text{std}}$) in Figure 5.6. The left panel replicates results of Lee *et al.* (2011) and shows that for large data sets, for which $\hat{\sigma}_{\text{std}}$ is small, the pragmatic Bayesian method produces much wider error bars than the standard method; $\hat{\sigma}_{\text{pB}}$ accounts for systematic and statistical errors, whereas $\hat{\sigma}_{\text{std}}$ only accounts for statistical errors. For the largest datasets $\hat{\sigma}_{\text{pB}}$ is twice as big as $\hat{\sigma}_{\text{std}}$. The right panel of Figures 5.6 shows that by converting systematic errors into statistical errors, the fully Bayesian method produces error bars more in line with $\hat{\sigma}_{\text{std}}$; although for the largest data sets $\hat{\sigma}_{\text{fB}}$ is bigger than $\hat{\sigma}_{\text{std}}$, it is not as big as $\hat{\sigma}_{\text{pB}}$.

Figure 5.7 compares the 1σ intervals for Γ produced by the three methods. Consider an interval computed under the pragmatic Bayesian method: $\mathcal{CI}_{\text{pB}}(\Gamma) = \{\hat{\Gamma} \pm \hat{\sigma}_{\text{pB}}(\Gamma)\}$, where $\hat{\Gamma}$ is the estimate of Γ under this method. To compare this interval with that computed under the standard method, we subtract $\hat{\Gamma}_{\text{std}}$ from $\mathcal{CI}_{\text{pB}}(\Gamma)$ and divide by $\hat{\sigma}_{\text{std}}(\Gamma)$. The result is an interval that extends from -1 to 1 if $\mathcal{CI}_{\text{pB}}(\Gamma)$ and $\mathcal{CI}_{\text{std}}(\Gamma)$ are identical, is wider if $\hat{\sigma}_{\text{pB}} > \hat{\sigma}_{\text{std}}$, and shifts to the left or right if $\hat{\Gamma}_{\text{pB}}$ and $\hat{\Gamma}_{\text{std}}$ differ. These adjusted intervals are plotted in the left panel of Figure 5.7. There is very little shifting to the left or right because the pragmatic Bayesian and standard methods produce very similar estimates. For small data sets (large values of $\hat{\sigma}_{\text{std}}$) the adjusted intervals are nearly $\{-1, 1\}$, but for

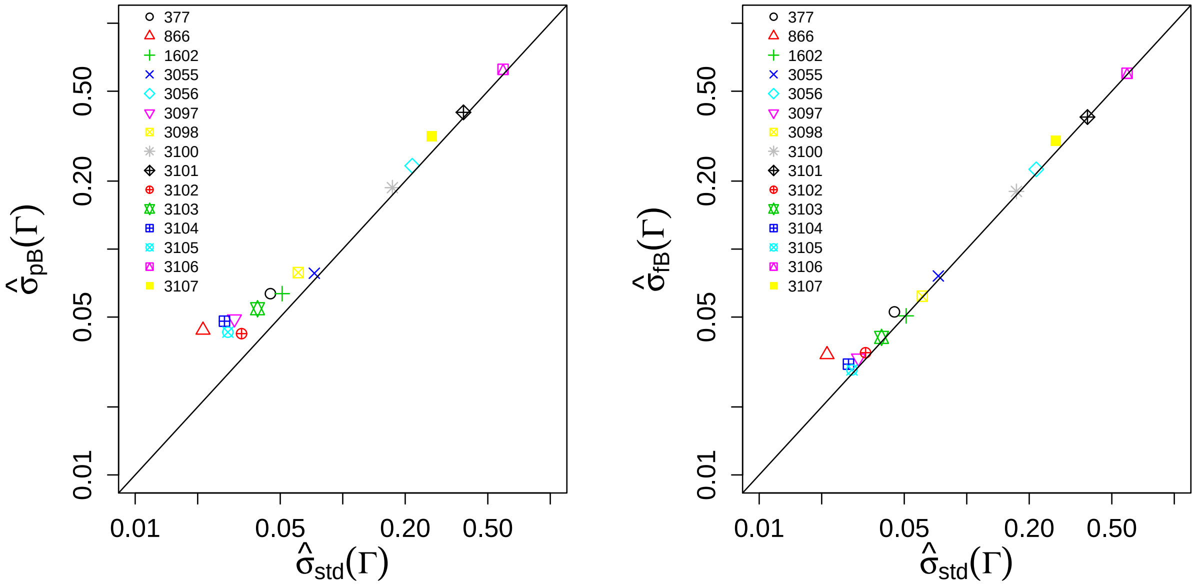

Figure 5.6: Comparing the error bars for Γ computed under the standard ($\hat{\sigma}_{\rm std}$), pragmatic Bayesian ($\hat{\sigma}_{\rm pB}$), and fully Bayesian ($\hat{\sigma}_{\rm fB}$) methods using spectra from each of fifteen radio-loud quasars. Smaller values of $\hat{\sigma}_{\rm std}$ correspond to data sets with more counts. For high-count spectra, $\hat{\sigma}_{\rm pB}$ tends to be substantially larger than $\hat{\sigma}_{\rm std}$ while $\hat{\sigma}_{\rm fB}$ is only moderately larger than $\hat{\sigma}_{\rm std}$. Thus, the fully Bayesian methods is able to provide a principled accounting for calibration uncertainty with only a moderate increase in the final error bars.

larger data sets (small values of $\hat{\sigma}_{\rm std}$) the adjusted intervals are as much as twice as wide. The right panel of Figure 5.7 adjusted intervals computed under the fully Bayesian method, $\mathcal{CI}_{\rm fB}(\Gamma)$, in the same manner. For the smaller data sets $\mathcal{CI}_{\rm fB}(\Gamma)$ and $\mathcal{CI}_{\rm std}(\Gamma)$ are similar, but for large data sets, they differ: the adjusted intervals tend to shift to the left or right but are not generally much more than four units wide. This means that the fully Bayesian method tends to adjust the fitted value and to increase error bars only moderately. In two cases (ObsID 3097 and 866) the fully Bayesian method shifts the fitted value of Γ by more than error bar ($\hat{\sigma}_{\rm std}$). This constitutes a significant shift in the scientific inference for these observations when calibration uncertainty is accounted for in a principled Bayesian manner.

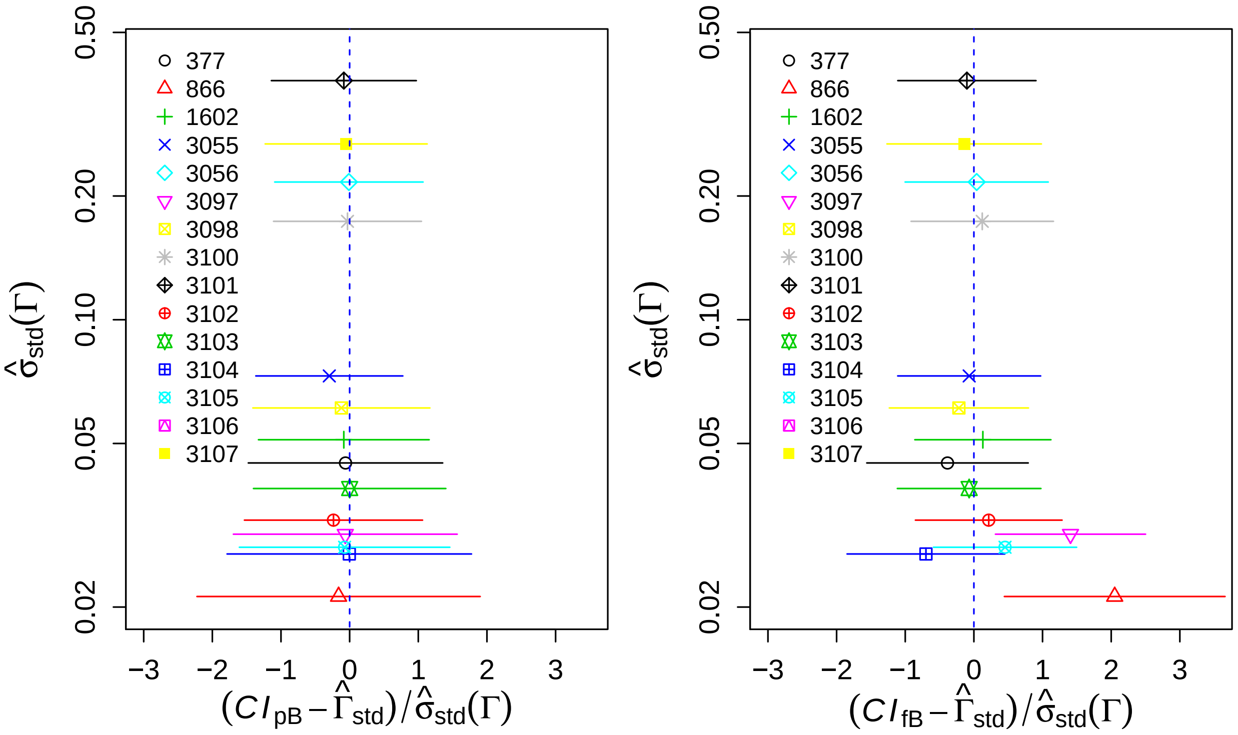

Figure 5.7: Comparing the confidence intervals for Γ computed under the standard, pragmatic Bayesian, and fully Bayesian methods using spectra from each of fifteen radio-loud quasars. The left panel plots 1σ confidence intervals computed with the pragmatic Bayesian method, but re-centered and re-scaled using the estimate and error bars of Γ computed under the standard method: $(\mathcal{CI}_{\rm pB} - \hat{\Gamma}_{\rm std})/\hat{\sigma}_{\rm std}(\Gamma)$. If the pragmatic and standard methods return the same estimates and error bars, the plotted intervals would equal the interval $(-1, 1)$. In fact the plotted intervals are as much as twice this wide for large-count (small $\hat{\sigma}_{\rm std}$) datasets, indicating that $\hat{\sigma}_{\rm pB}$ can be substantially larger than $\hat{\sigma}_{\rm std}$. The right panel plots 1σ confidence intervals computed with the fully Bayesian method, re-centered and re-scaled in the same manner. The plotted intervals shift right and left because the fitted values under the standard and fully Bayesian methods differ. The width of the fully Bayesian intervals, however, are only moderately larger than under the standard method.

5.4 Fitting a multi-thermal spectral model

Our analyses thus far have been carried out using a simple power-law spectral model, for both simulations (Section 5.1) and observed data (Section 5.2). Here we illustrate the pragmatic and fully Bayesian methods using a more complex multi-component thermal model.

We choose one of the strongest sources in the Chandra Source Catalog, ζ Ori, a young (<12 Myr) binary system comprised of an O9 supergiant that is X-ray bright, and a weaker B0 subgiant about $3''$ away. The source is observed (ObsID 1878) at 15 arcmin off-axis, and situated on the ACIS-S2 chip, and is detected with a count rate of 1.33 ct s^{-1}, with $> 10^5$ counts in 75.46 ksec. Because of the large off-axis location, the point spread function is broad, and the binary cannot be spatially resolved. Furthermore, the source is spread out over > 20000 pixels, with maximum fluence at < 0.0017 ct s^{-1} at any pixel, and CCD pileup effects may be ignored.

Our objective here is not to model the spectrum in detail (the X-ray emission is thermal, generated from shocked plasma deep in the wind; see Waldron and Cassinelli (2001), Pollock (2007), Raassen *et al.* (2008), Herve *et al.* (2013) for various models designed to account for X-ray emission from massive stars); rather, it is to consider the effect calibration uncertainty has on spectral fitting. We construct a variable-abundance absorbed 2-temperature APEC spectral model and fit it to the data. This mimics roughly previous attempts to model spectra of ζ Ori obtained with other telescopes such as ASCA (Yamauchi *et al.*, 2000) and XMM-*Newton* (Raassen *et al.*, 2008). For reference, Yamauchi *et al.* find two temperature components at $T_1 = 0.2$, $T_2 = 0.6$ keV, at an absorption column fixed at $N_H = 2.6\,10^{20}$ cm^{-2}; Raassen *et al.* find three temperature components at $T_1 = 0.55$, $T_2 = 0.2$, $T_3 = 0.07$ keV, $N_H = 5\,10^{20}$ cm^{-2}, with abundances of C, N, and O, being close to solar photospheric (represented by the compilation of Anders and Grevesse (1989)), and those of Ne, Mg, Si, and Fe being elevated. The results of applying the standard analysis, pragmatic Bayesian, and fully

Bayesian methods to the data are given in Table 5.3. We find significantly higher absorption columns, and lower temperatures and abundances, with our estimated abundance consistent with the low metallicities measured for the nearby NGC 2023 star cluster (Lpez-Garca *et al.*, 2013). We find that these results are stable with respect to calibration uncertainty, with all three methods producing similar best-fit values.

The fits are also summarized in Figures 5.8 and 5.9. In particular, they show that relative to the standard method, the pragmatic Bayesian method delivers similar fitted values for T_1 and T_2, but accounts for calibration uncertainty by inflating their error bars. The fully Bayesian method shifts the fitted values by a small amount and generally requires a smaller increase in the error bars. In this case, the posterior correlation of T_1 and T_2 decreases under the fully Bayesian method, see Figure 5.9.

Table 5.3: Fit parameters for ζ Ori

Model	Standard		Pragmatic Bayesian		Fully Bayesian	
Parameters	Analysis		Analysis		Analysis	
N_H $[10^{22}\,\mathrm{cm}^{-2}]$	0.1475	±0.00901	0.1498	±0.00916	0.1511	±0.00866
T_1 [keV]	0.179	±0.0016	0.179	±0.0022	0.181	±0.0019
T_2 [keV]	0.475	±0.0063	0.474	±0.0069	0.471	±0.0068
$[C, N, O]^a$	0.23	±0.018	0.23	±0.024	0.21	±0.026
$[Ne]^a$	0.48	±0.031	0.48	±0.033	0.47	±0.034
$[Ni, Mg, Si, Ca, Fe]^a$	0.41	±0.028	0.41	±0.032	0.40	±0.032
Norm_1 $[10^{14}\ \mathrm{cm}^{-5}]$	0.0628	±0.0065	0.0640	±0.0078	0.0588	±0.0056
Norm_2 $[10^{14}\ \mathrm{cm}^{-5}]$	0.0105	±0.00057	0.0107	±0.00079	0.0100	±0.00059

a : abundances relative to solar (Anders and Grevesse, 1989)

The second row of Figure 5.8 shows time series plots of the Markov chains of T_1 used to simulate the three posterior distributions. While all three converge reasonably well, the fully Bayesian chain occasionally "sticks" at a particular value of the parameter for a number of iterations. This indicates that the pragmatic proposal distribution may attribute relatively little probability to some regions of the parameter space with appreciable probability under $p_{\mathrm{fB}}(\theta, A \mid Y)$. This can also be seen in Figure 5.9 where the 90% contours of (normal approximations to) the pragmatic and fully Bayesian posterior distributions are plotted in

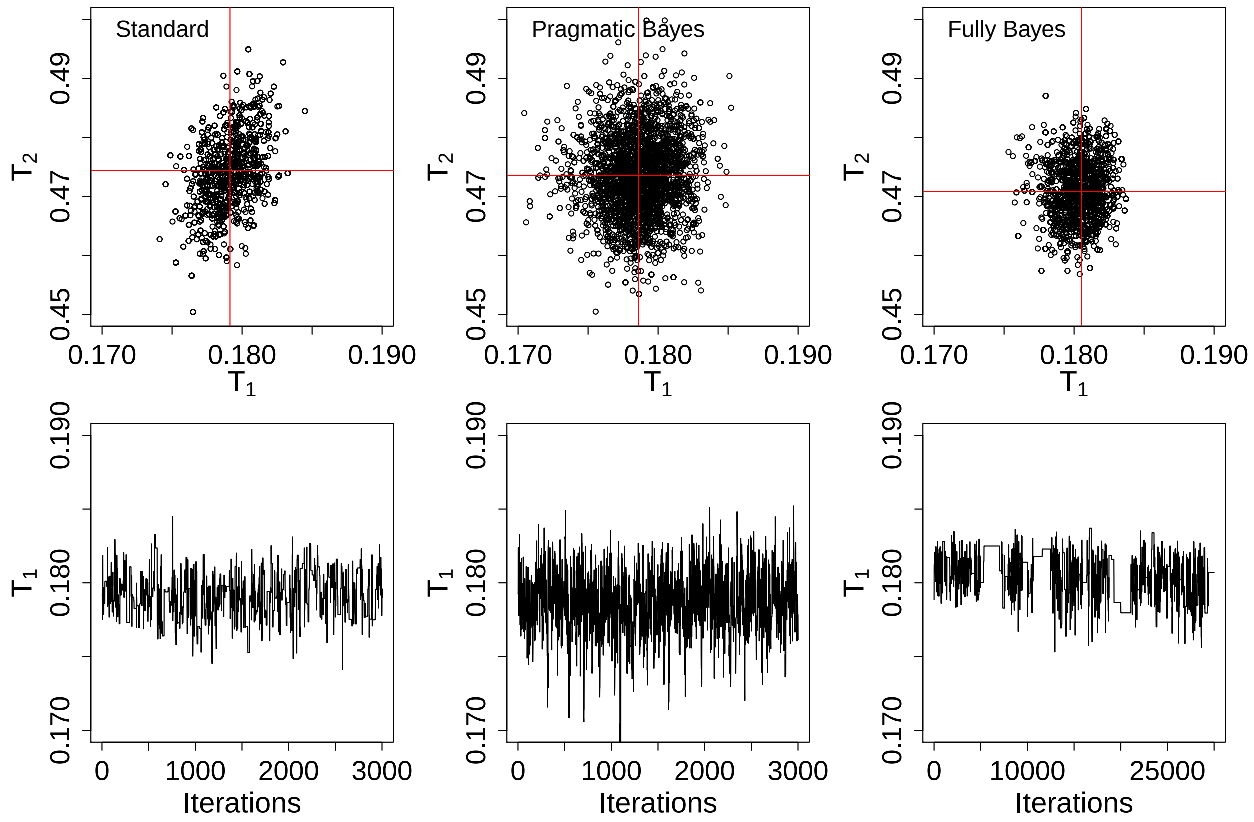

Figure 5.8: The standard, pragmatic Bayesian, and fully Bayesian fits of the T_1 and T_2 parameter in the spectrum of ζ Ori. The first row plots a Monte Carlo simulation from each of the three posterior distributions; the solid red lines correspond to the posterior means of T_1 and T_1 under each fit. The fitted values (posterior means) for the two parameters are indistinguishable under the standard and pragmatic Bayesian methods, but shift noticeably under the fully Bayesian method. Error bars are quantified by the spread of the simulated parameter values under each of the methods. The error bars under the pragmatic Bayesian method are noticeably larger than those computed with the other two methods. Thus, the fully Bayesian method again is able to account for calibration uncertainty by shifting the fitted values rather than by increasing their error bars. The second row presents time series plots of the Markov chains for T_1 used to generate the three Monte Carlo simulations. While all three chains are fairly well behaved, the fully Bayesian chain occasionally "sticks" at the same parameter value for a number of iterations. This is an indication that the pragmatic proposal distribution attributes relatively little probability to some regions of the parameter space with appreciable probability under $p_{\mathrm{fB}}(\theta, A \mid Y)$. Nonetheless, the algorithm preforms well enough for valid inference.

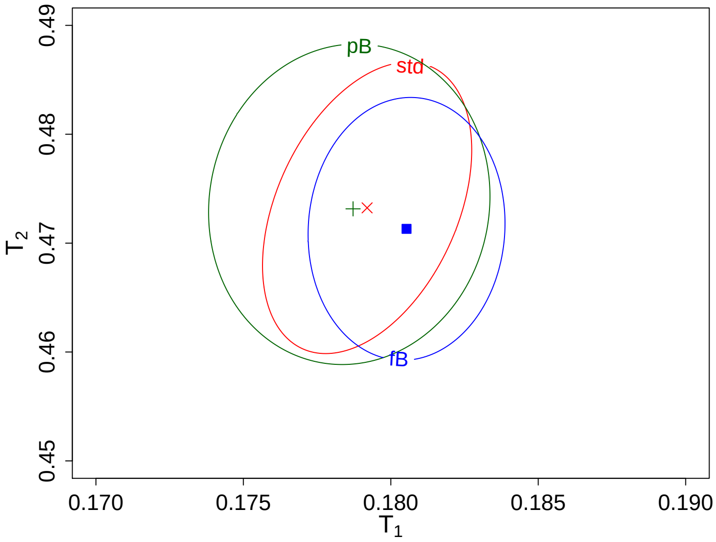

Figure 5.9: The plot superimposes the 90% contours of the three posterior distributions simulated in the first row of Figure 5.8. The contours are computed using a normal approximation to the posterior distribution computed under the standard (red), pragmatic Bayesian (green), and fully Bayesian (blue) methods. The posterior means under the three methods are plotted, respectively, as a red ×, a green +, and a filled blue square. Relative to the standard method, the pragmatic Bayesian method delivers similar fitted values and larger error bars, while the fully Bayesian method shifts the fitted value but only moderately inflates the error bars. In this example the correlation of the two parameters decreases under the fully Bayesian method.

green and blue, respectively. The fact that the fully Bayesian contour extends outside the pragmatic Bayesian contour indicates that we may have trouble exploring parts of $p_{\text{fB}}(\theta, A \mid Y)$ using the pragmatic proposal distribution. (Recall that we require the jumping rule of the independence sampler to be an over-dispersed approximation to $p_{\text{fB}}(\theta, A \mid Y)$.) The mixed pragmatic proposal sampler, however, mixes the pragmatic proposal with a random walk update. This second component allows simulation of parameter values with relatively low probability under the pragmatic proposal rule.

This, however, comes at significant computational cost. Owing to its "sticking" we ran the fully Bayesian sampler for 30,000 iterations. (The pragmatic sampler was run for 3000.) This combined with complexity of the multi-thermal spectral model fit with its eight parameters resulted in a significant computation burden. The analyses in this section required about one week of computing time compared with about an hour for the blackbody model that we will describe in Section 5.5 and about 90 minutes for each of the quasars fit in Section 5.3.

In the fully Bayesian run, the data prove to be informative about the effective areas. The subset of $\mathcal{A}$ that is consistent with the data and the adopted model suggests that the nominal effective area is underestimated (see Figure 5.10) This is not surprising, since effective areas at large off-axis angles are not as well calibrated as near the aimpoint. Our analysis suggests that at off-axis angles > 15 arcmin the *Chandra* effective areas must be increased by ≈10%.

5.5 Fitting a blackbody model to a grating spectrum

As an additional example of the versatility of our approach, we consider data from an entirely different detector, fitted with a blackbody spectral model. We analyze a high-resolution grating spectrum of an isolated Neutron Star, RX J1856.5-3754 (RXJ1856). This source has been observed with the LETGS+HRC-S grating/detector combination numerous

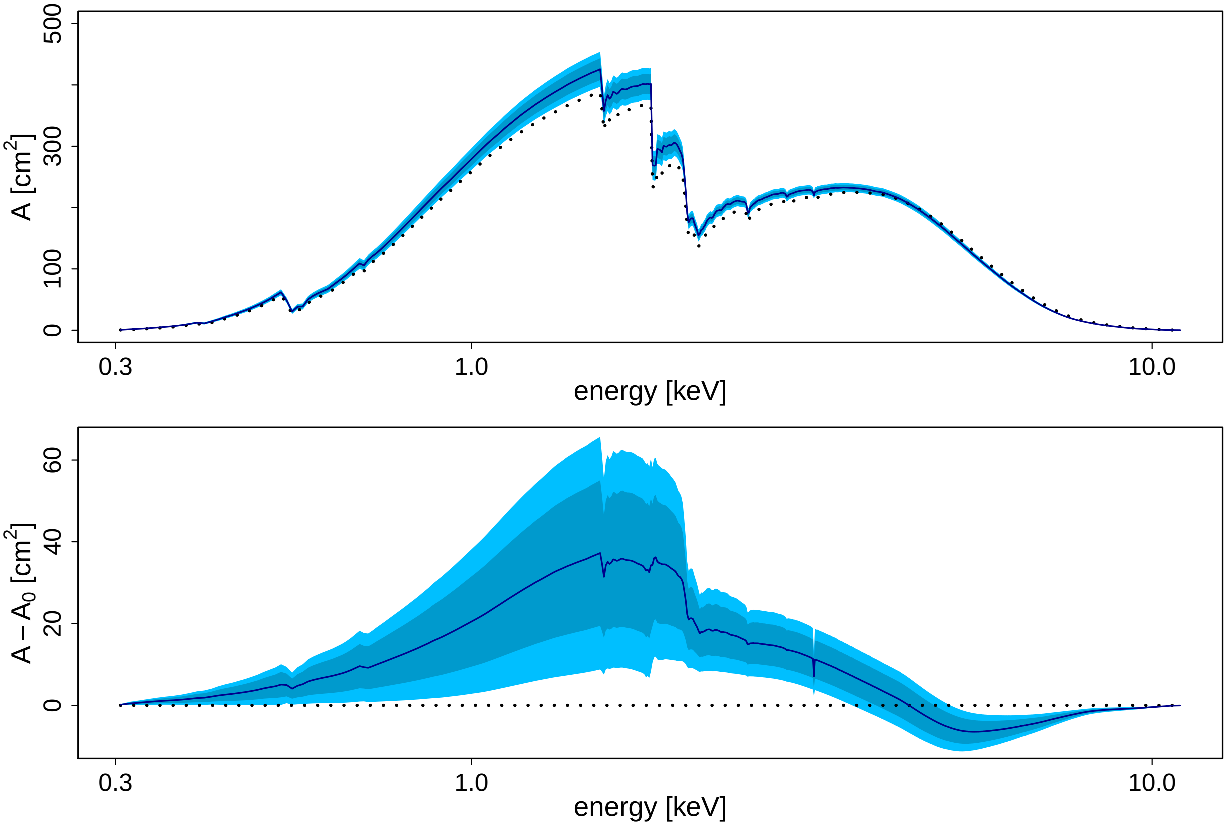

Figure 5.10: Estimating the range of effective area curves that are consistent with the spectrum of ζ Ori. The plots summarize $p_{\mathrm{fB}}(A \mid Y)$. The pointwise posterior distribution of A is plotted in blue, where the dark (light) blue area corresponds to the central 68.3% (90%) region of $p_{\mathrm{fB}}(A \mid Y)$. The pointwise posterior mean of A and its default value, A_0 are plotted as solid blue and dotted black lines, respectively. The top panel shows the full effective areas and the bottom panel subtracts off the default effective area curve, A_0, to highlight the differences. (We cannot subtract off the true curve as in Figure 5.3 because it is unknown.) The data suggest that A_0 underestimates the true effective area by ≈10%, and the vignetting correction must be reduced for large off-axis angles for *Chandra* imaging observations.

times over the *Chandra* mission, and has accumulated 617.735 ks of exposure. RXJ1856 is intrinsically an interesting object: it was originally classified as an isolated Neutron Star, but was later suspected to be a Quark Star Drake *et al.* (2002). The X-ray data were fit as a blackbody spectrum with a temperature of $T = 61.1 \pm 0.3$ eV by Drake *et al.*, resulting in a radius estimation of ≈4-8 km. But the optical data are inconsistent with the X-ray predictions, and require fitting by a more complex magnetic Hydrogen atmosphere model, with a temperature of $T_\infty \approx 37$ eV and a conventional radius $R_\infty \approx 17$ km consistent with a neutron star core Ho *et al.* (2007). In the X-ray regime itself, these two models cannot be statistically distinguished (Ho *et al.* (2007)).

Table 5.4: Fit parameters for RX J1856.5-3754

Model	Standard		Pragmatic		Fully	
Parameters	Analysis			Bayesian		
N_H [10^{22} cm^{-2}]	0.0091	±0.00043	0.0091	±0.00058	0.0093	±0.00055
T [eV]	62.4	±0.58	62.5	±1.05	62.4	±0.93
Norm [0.083 ergs s^{-1} cm^{-2}]	0.00031	$\pm 5.6 \times 10^{-6}$	0.00031	$\pm 2.3 \times 10^{-5}$	0.00032	$\pm 2.2 \times 10^{-5}$
Background scale	65.6	±1.5	65.8	±1.8	65.5	±1.7

Here, for the sake of simplicity, we adopt an absorpbed Blackbody spectrum model to fit the soft X-ray data. This spectral model was fit using exactly the same methods and algorithms as in Section 5.3, except that the background was modeled as a fixed 8^{th}-degree polynomial whose coefficients were determined via a standard fit to the background spectrum, and was then incorporated into the source model with a variable normalization.

As was done for the *Chandra*/ACIS-S, we generate a calibration library based on constrained spline curve modifications of known subsystem uncertainties in the LETGS+HRC-S system (Drake et al., in preparation). We have computed the principal components for this library, and show the top 5 components (in color), which together account for $> 95\%$ of the variance in the library, in Figure 5.11. Also shown, in grey, are the summed contributions of the remaining 5% of the components. Notice that there are wavelength regions where this residual could be a significant factor. The wavelength range over which the RXJ1856 data are informative are shown with vertical dashed lines, and residual components do not affect the analysis over this range. Note that in all our analyses below, we use 8 principal components.

We limit our analysis to the wavelength ranges $[+25 : +59.5, +68 : +80]$ Å, with the gap centered on the HRC-S chip gap. The chip gap is excluded because, even though the nominal effective area A_0 includes the effect of dither and corrects for the drop across the gap, it is subject to additional systematic errors due to deformations in where the active regions on the chip are located, and these are not included in the calibration library. We co-add the spectra from all positive dispersion datasets of RXJ1856. We exclude the negative dispersion data because it has a chip gap at a different wavelength range, which we make it difficult to interpret the analysis results. There are ≈129 kct in the spectrum over the chosen wavelength range, of which ≈43.8 kct are estimated to be due to the background. Though the large number of net counts (≈85 kct) puts this dataset well in the range of that likely to be affected by calibration uncertainty, the large fraction of expected background counts makes this an inefficient dataset to place constraints on the calibration library. As expected, the application of pragmatic Bayesian method expands the error bars on the best-fit model parameters (see Table 5.4), For instance, the temperature estimate remains stable at $T = 62.4$ eV, but the uncertainty increases from ±0.6 eV for the standard analysis to ±1.05 eV for pragmatic Bayesian, and decreases slightly to ±0.93 eV for the fully Bayesian analysis (see Figure 5.12).

Unlike the case with ζ Ori (Section 5.4), there is no significant effect on the range of effective area curves in the calibration library that are consistent with the spectrum of RXJ1856 (see Figure 5.13). We attribute this lack of sensitivity partly to the high background that contaminates the dataset, and also to the short wavelength range over which the dataset is informative. As we see from the Principal Components displayed in Figure 5.11, there are long-range correlations present in the library which will be selected when a source with a suitably long wavelength range is analyzed. In the current analysis, there is a suggestive small curvature bias of ∼1-2% in the HRC-S/LETG effective area over the $25 - 80$ Å range.

This bias, however, is fully contained within the nominal 1σ range of the effective area curves.

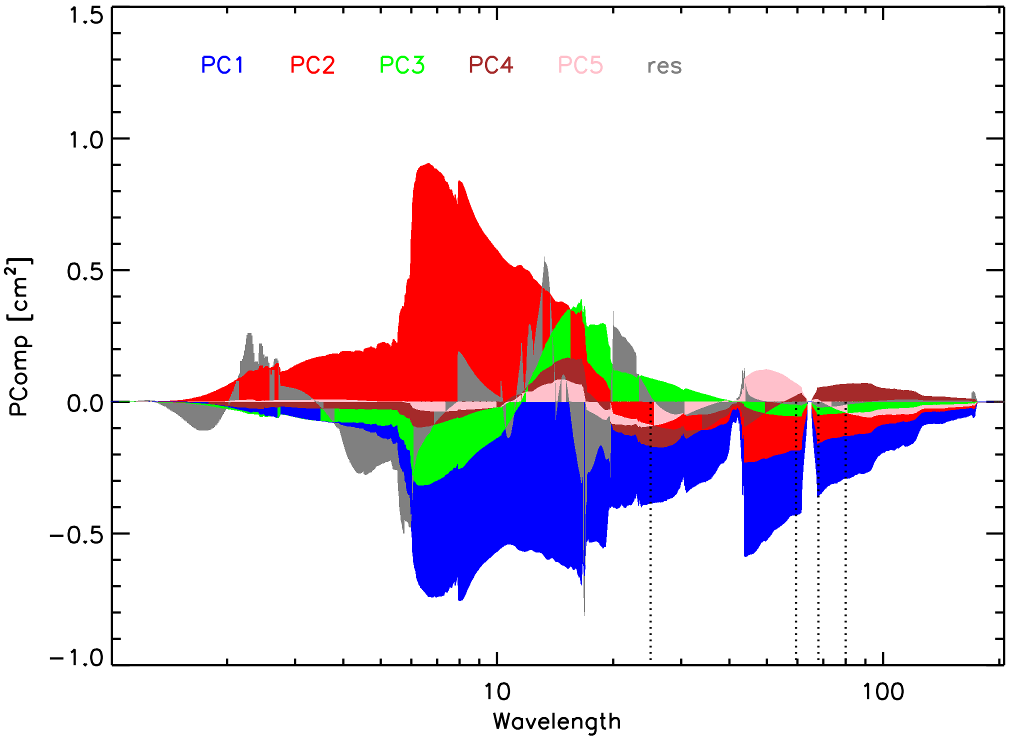

Figure 5.11: Top 5 Principal Components of the *Chandra* LETGS+HRC-S effective area calibration library. The 1st (blue), 2nd (red), 3rd (green), 4th (brown), and 5th (pink) components, weighted by the square-root of their eigenvalues, are shown as colored regions. The the sum of the similarly weighted contributions from the remaining components is shown as grey. The vertical dashed lines indicate the wavelength range over which the data analysis is carried out.

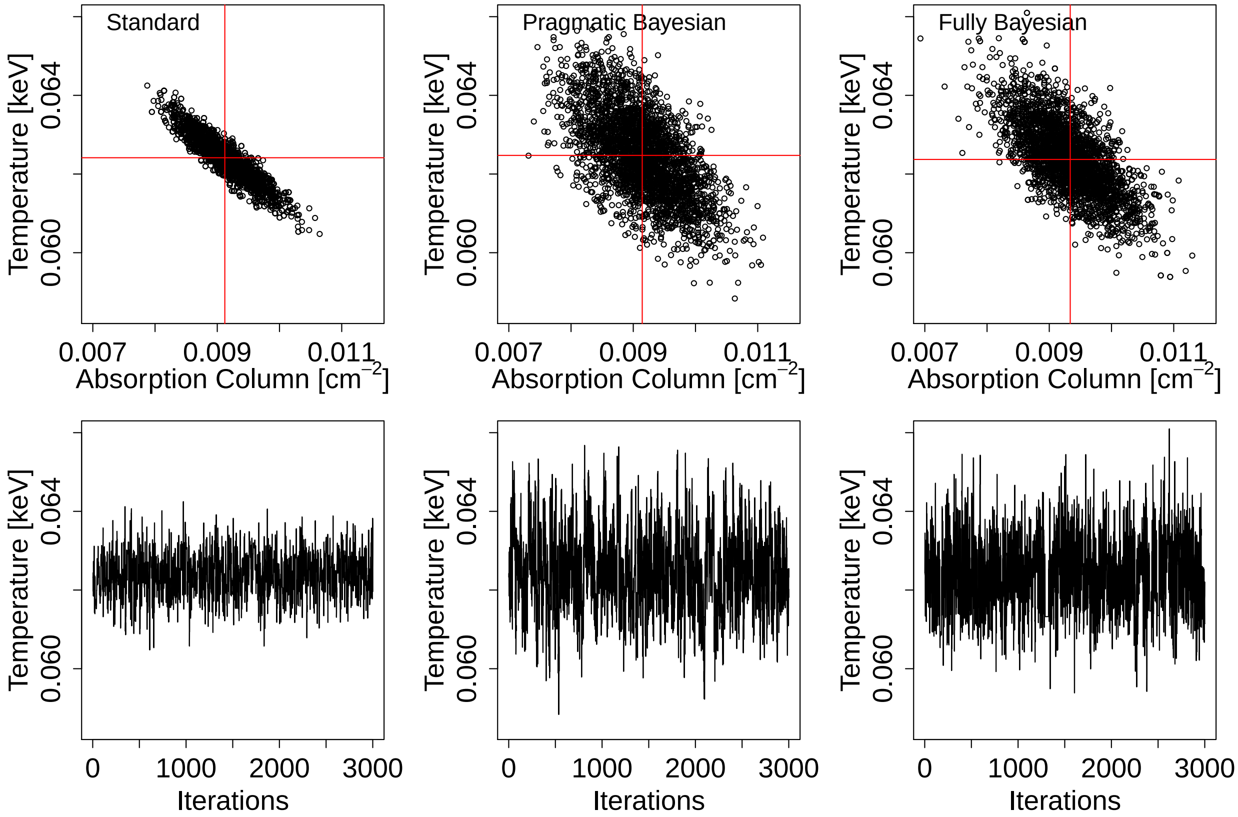

Figure 5.12: The standard, pragmatic Bayesian, and fully Bayesian fits of the the temperature and absorption column in the spectrum of RX J1856.5-3754. The plots are layered as in Figure 5.8. In this case, however, there is only a small shift of the fitted values (posterior means given by the solid red lines). There is a noticeably less joint uncertainty in the two parameters under the fully Bayesian fit than under the pragmatic fit. There is no noticeable sticking for the any of the MCMC samplers; compare the second row with that of Figure 5.8.

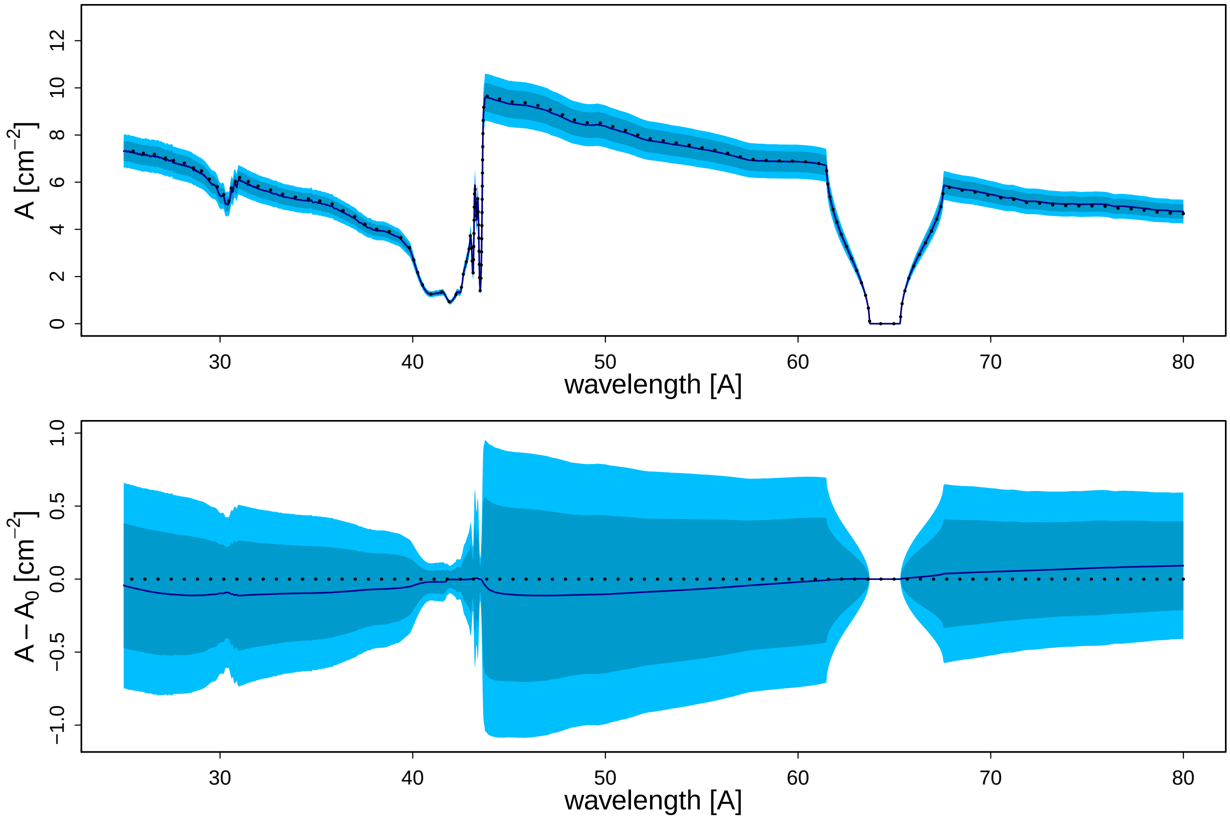

Figure 5.13: Estimating the range of effective area curves that are consistent with the spectrum of RX J1856.5-3754. The plots summarize $p_{\rm fB}(A \mid Y)$. The pointwise posterior distribution of A is plotted in blue, where the dark (light) blue area corresponds to the central 68.3% (90%) region of $p_{\rm fB}(A \mid Y)$. The pointwise posterior mean of A and its default value, A_0 are plotted as solid blue and dotted black lines, respectively. The top panel displays the the effective areas and the bottom panel shows the same data with the default effective area, A_0, subtracted off to highlight the differences. (We cannot subtract off the true curve as in Figure 5.3 because it is unknown.) Here the fitted (posterior mean) and default (dotted black curve) are very similar.

Chapter 6

Uncertainty in Photon Redistribution Matrix

In this chapter, we will focus on another source of calibration uncertainty, the photon redistribution matrix. Unlike the procedure used to analyze uncertainty in the effective area curve, here we will first summarize the photo redistribution matrix by a normal approximation for each row, and then conduct PCA to the normal density parameters. With this summarization, the standard, pragmatic Bayesian and fully Bayesian methods are applied to account for uncertainty in the photon redistribution matrix in spectral analysis. Because of the complexity of the photon redistribution matrix, incorporating its uncertainty becomes much more challenging. We illustrate that the same sampling algorithms apply as in the case of the effective area curve, and the simulation results demonstrate that the advantages of the fully Bayesian method preserves.

6.1 Summarizing the Uncertainty of Photon Redistribution Matrix

In this section, we describe the concept of the photon redistribution matrix, and provide the reasons why simple PCA cannot be used to summarize the redistribution matrix. We will develop a more complicated algorithm in Section 6.1.3 to capture the uncertainty of the photon redistribution matrix.

6.1.1 Photon Redistribution Matrix

Since the detectors do not have perfect resolution, the recorded energy of a photon may be different from its true energy. And this spreading of the energy is typically represented by a matrix, the photon redistribution matrix. In high resolution instruments (e.g. diffraction gratings, such as HETG (High Energy Transmission Grating) and LETG (Low Energy Transmission Grating)) the matrix is almost diagonal. In proportional counters the matrix elements are non-zero over a large area. CCD (Charge-Coupled Device) detectors, such as ACIS (Advanced CCD Imaging Spectrometer), are an intermediate case, with most of the response being almost diagonal, but escape peaks and low energy tails adding significant contributions.

Photon redistribution matrix element, $R(E^*; E)$, represents the probability that an incoming photon with energy E will be detected in the output detector channel E^*. We represent $R = \{R(E^*; E)\}$ as the whole photon redistribution matrix, with the dimension $N \times K$, where N is the number of the energy bins and K is the number of channel bins of the detector. Each row n of R can be regarded as a probability density function, thus the sum of each row equals one. Figure 6.1 shows one example of the photon redistribution matrix of an ACIS detector. To better explore the structure of this matrix, we use a color-scheme

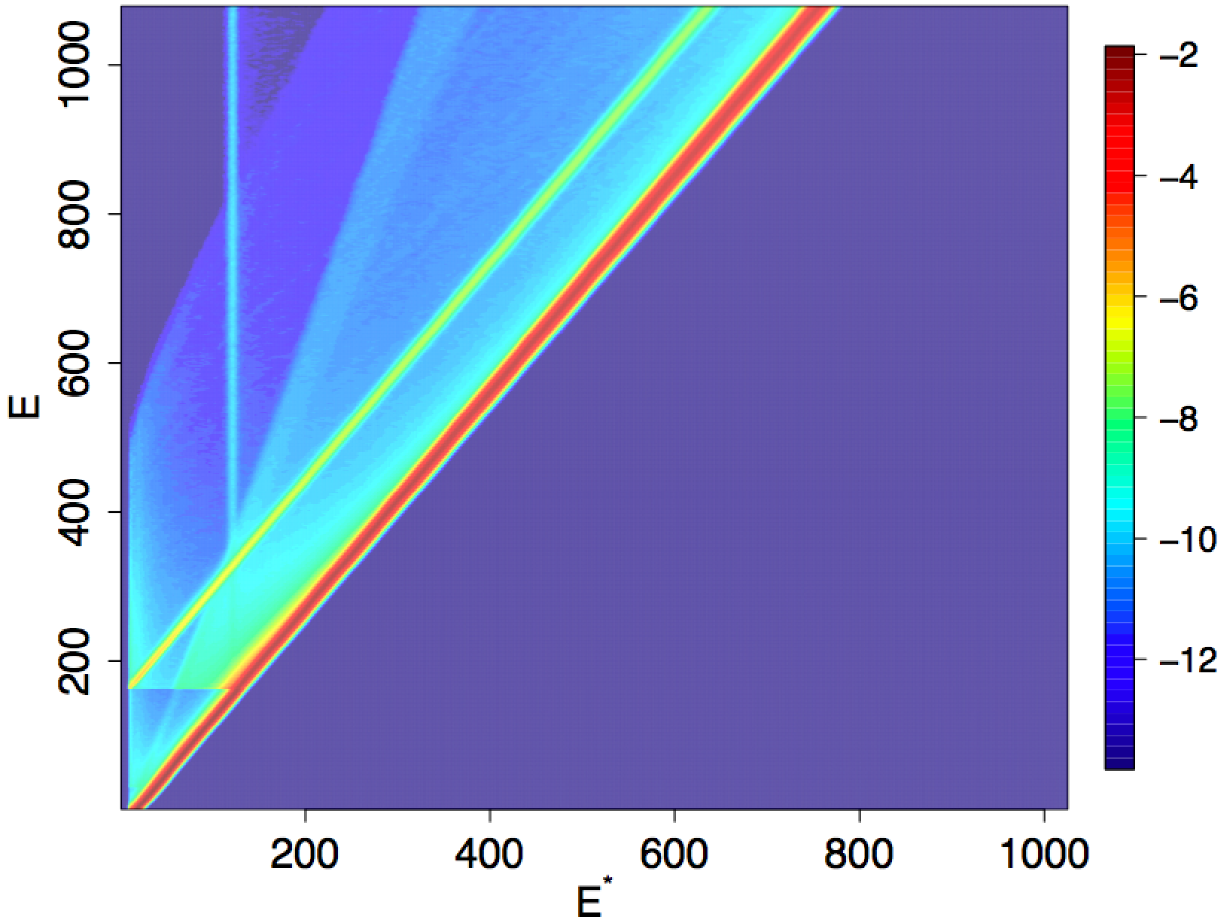

Figure 6.1: One example of photon redistribution matrix. This image is the plot of logarithm transformation of R, $log(R + 10^{-6})$. It clearly shows that most matrix elements are equal or close to zero. The main probability lies on two diagonal lines, and the right line takes about 98% of the photon response.

based on the logarithm transformation of R, $log(R + 10^{-6})$. This redistribution matrix has the dimension 1078×1024. The image clearly shows that most matrix elements are equal or close to zero. Most of the probability lies on two diagonal lines, and the main diagonal line takes about 98% of the photon response. In reality, since lots of the matrix elements are zero, photon redistribution matrix is stored in a standard FITS (Flexible Image Transport System) file format to save hardware space. FITS file format only stores the value of the nonzero elements of the matrix, and the index of the these values.

As in the effective area curve case, a library of ACIS redistribution matrices is generated to represent the best quantification of the calibration uncertainty in redistribution matrix. In our general notation, we denote the redistribution matrix library by $\mathcal{R} = \{R_1, R_2, ..., R_L\}$, where $L = 1000$, and let R_0 denote the default redistribution matrix associated with the library. For this library, all the matrices have the same dimension, $M = 1078$ and $K = 1024$.

6.1.2 Naive PCA Summary

In order to summarize the redistribution matrix into a concise and usable form, we need to reduce the dimension from more than one million (1078×1024) into a feasible number and then set up a proper prior distribution $\pi(R)$ for R. Thus, we can sample a new redistribution matrix R_{new} from this prior distribution and evaluate the prior probability for R_{new}. Hence, the pragmatic Bayesian and fully Bayesian methods can be applied.

Similar to the PCA for the effective area curve, we have the following method to summarize the redistribution matrix.

Naive PCA Summary:

Step 1: vectorize the redistribution matrix R into $\tilde{R}$

Step 2: perform PCA summarization for L one-million-dimensional vectors, $\tilde{R}$ (see the detail algorithm in Section 1.5)

Step 3: use the PCA result to sample new $\tilde{R}_{new}$ (similar to the case in Section 2.3)

Step 4: transform the vector $\tilde{R}_{new}$ into matrix format, R_{new}

There are several issues with this easy algorithm. First, to perform PCA for L million-dimensional vectors requires a large amount of computation. Secondly, this algorithm ignores the inherent constraints of the redistribution matrix. Since each row of the redistribution matrix represents a probability density function, all the elements should be bigger than or equal to zero and each row sums to one. This algorithm cannot guarantee that the new redistribution matrix has these properties. Lastly, when we use the PCA result to formulate the prior distribution $\pi(R)$, we essentially assume that the uncertainty in R can be described by a multivariate normal distribution; see Section 1.5. Figure 6.2 shows the histogram of 25 selected matrix elements from the L redistribution matrix samples. These 25 elements are along the main diagonal line of Figure 6.1, and are relatively large probabilities. It demonstrates the probability density of each of these 25 elements is highly left-skewed and is nowhere close to the normal density curve. Thus, simply use this naive method to summarize the redistribution matrix in inappropriate.

6.1.3 Redistribution Matrix Summary

In order to better summarize the redistribution matrix, we explore the structure of each row of the redistribution matrix. In more detail, from Figure 6.1, we can see the main proportion of probability lies along the main diagonal, and it is reasonable to assume that the main uncertainty of the redistribution matrix comes from these elements near this main diagonal. We will focus on the summarization of these elements.

Figure 6.3 illustrates the probability density structure of each row of the default redistribution matrix, R_0. Each curve represents the probability density function of the redistribution of one photon in energy bin E, and E^* represents its recorded energy (channel bin). The number on top of the curve is the row indictor of the redistribution matrix, which is an index for E. From this figure, we can see the probability density curve for each row has the symmetry and bell-shaped properties. Notice that this curve is different from the density we describe in the Figure 6.2 in Section 6.1.2, which is based on only one matrix entry across 1000 redistribution matrix samples. This finding enables us to use normal approximation to

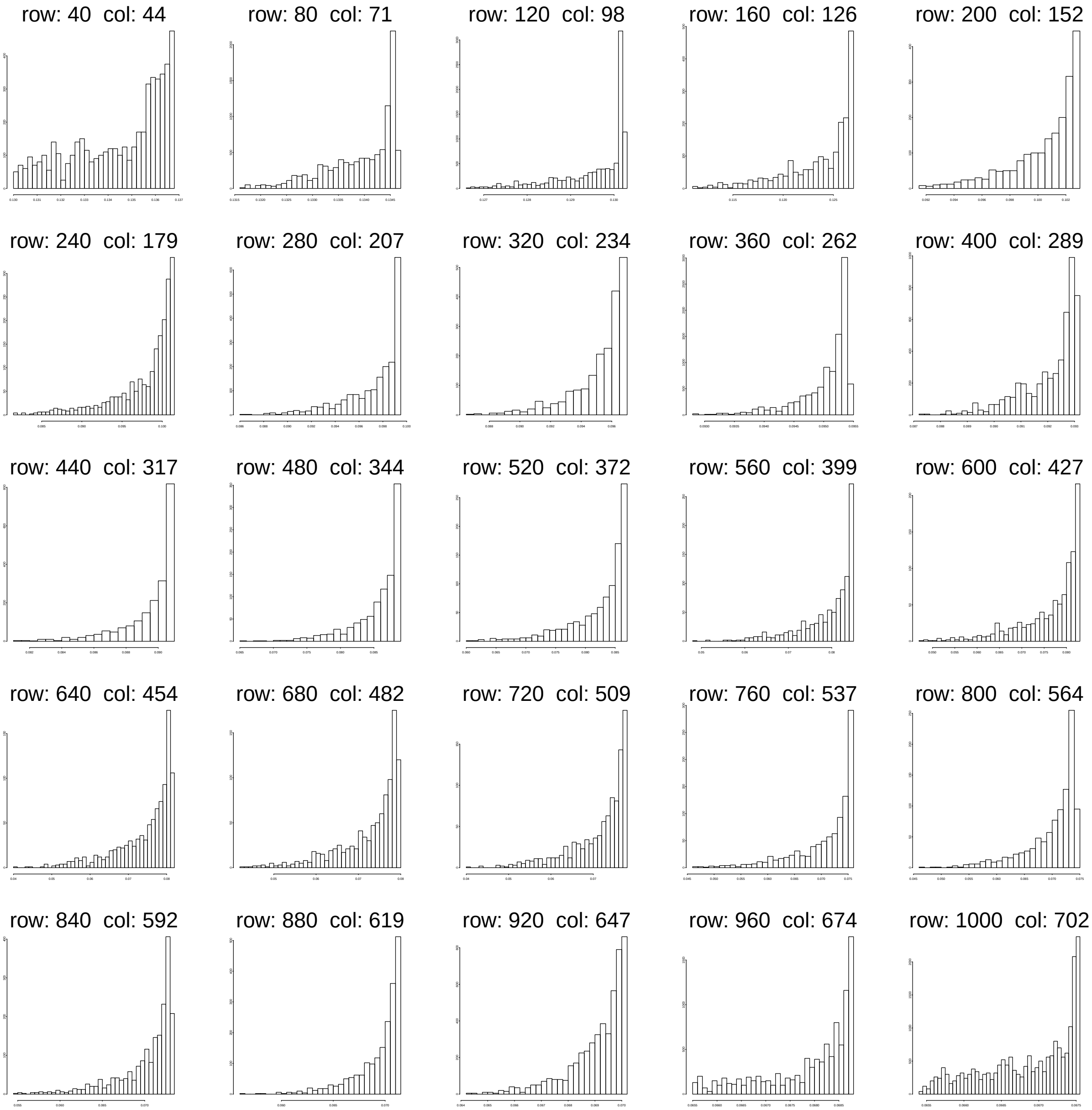

Figure 6.2: Histogram of 25 selected redistribution matrix elements. These 25 elements are along the main diagonal line of Figure 6.1, containing relatively large probabilities. It shows the probability density of these 25 elements is highly left skewed and far away from the normal distribution.

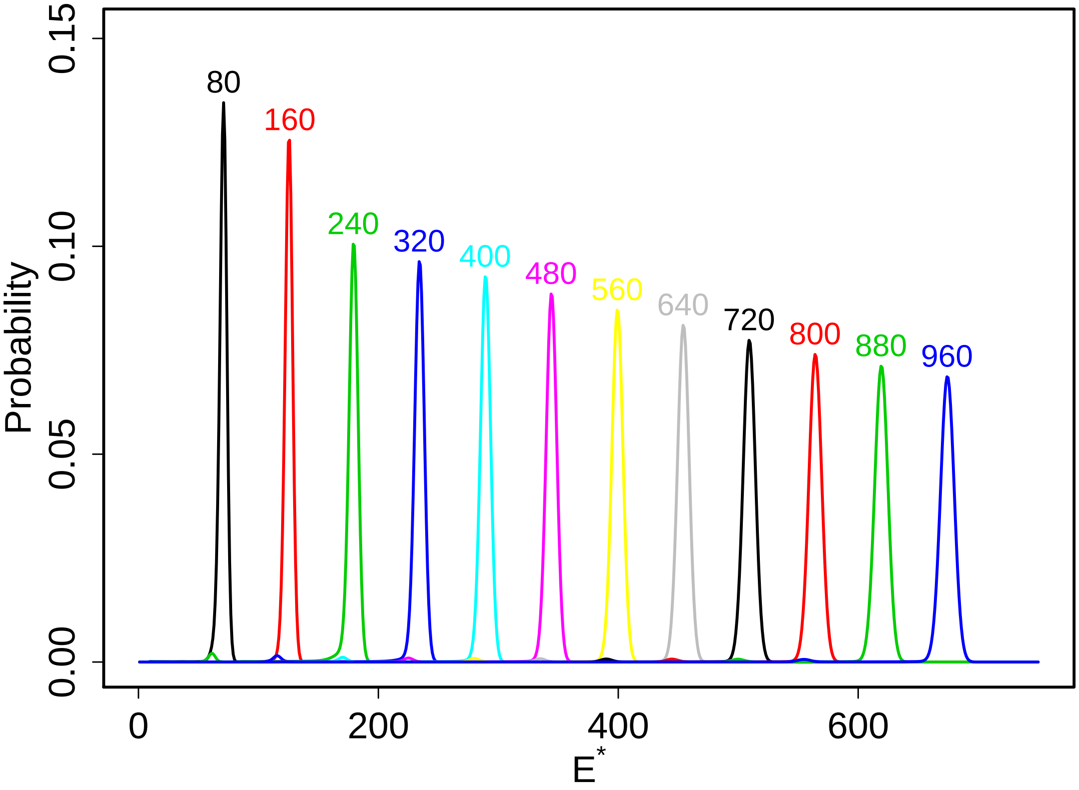

Figure 6.3: Probability density curve of each row of default redistribution matrix, R_0. Each curve represents the probability density function of the redistribution of one photon in energy bin E, and E^* represents its recorded energy (channel bin). The number on top of the curve is the row indictor of the redistribution matrix, which is an index for E. For example, the left black curve represents the density curve of the 80th row of R_0. It shows that the probability density curve for each row has the symmetry and bell-shaped properties, which enables us to use normal approximation to summarize the probability density.

fit each probability density curve, thus reduces the dimension of each row n from K (1024, number of channel bins) to 2 (mean and variance of approximated normal density function). In this way, we can efficiently compress a redistribution matrix R with the dimension $N \times K$ (1078×1024) into a parameter vector Ψ with dimension $2N$ ($1078 \times 2 = 2156$). Once L redistribution matrices in the library are compressed into L $2N$-dimensional vectors, PCA can be used to summarize the correlation structure of Ψ.

Here, we propose the algorithm for photon redistribution matrix summarization:

Redistribution Matrix Summarization:

Step 1: Each $R_i \in \mathcal{R}$ is separated into two matrices, $R_i = R_i^b + R_i^s$, where R_i^b only contains big elements along the main diagonal line (about 30-50 elements per row; other elements are zero), and R_i^s contains all other small values. We only consider the uncertainty in R_i^b. Figure 6.4 illustrates the boundaries that are used to separate R_i into R_i^b and R_i^s. The elements between the two black lines are considered as large value entires, and thus used to construct R_i^b. R_i^s is built with the other elements. The separation boundaries are the same for each R_i.

Step 2: For each row n of R_i^b, $R_i^b[n,]$, we approximate it with normal density function, with $\mu_{i;n} = \underset{k}{\operatorname{argmax}}\ R_i^b[n,k]$, and $\sigma_{i:n}^2 = \sum_{k=1}^{K} R_i^b[n,k](k - \mu_{i;n})^2$, where we set the mode location of $R_i^b[n,]$ as the mean of the normal density function.

Step 3: The normal approximation allows us to transform R_i^b into $2N$-dimension vector Ψ_i, $\Psi_i = \{(\mu_{i;1}, \sigma_{i:1}^2), ..., (\mu_{i;n}, \sigma_{i:n}^2), ..., (\mu_{i;N}, \sigma_{i:N}^2)\}$. The redistribution matrix library $\mathcal{R}$ is therefore written as $\ominus = \{\Psi_1, ..., \Psi_i, ..., \Psi_L\}$

Step 4: We conduct PCA for the mean subtracted $\ominus - \bar{\Psi} = \{\Psi_1 - \bar{\Psi}, ..., \Psi_i - \bar{\Psi}, ..., \Psi_L - \bar{\Psi}\}$, and obtain the biggest J eigenvalues $\{r_1^2, r_2^2, ..., r_J^2\}$ and the corresponding eigenvectors $\{v_1, v_2, ..., v_J\}$, where $r_1^2 \geq r_2^2 \geq, ..., \geq r_J^2$ as in Section 1.5

In Step 1, since we only focus on the uncertainty on the elements along the main diagonal line, we cut the redistribution matrix into two matrices and use the normal approximation for the important elements. Otherwise, using one simple normal distribution to approximate the whole row vector of redistribution matrix would not be appropriate because of the existence of a secondary diagonal line and other subtle structures in the redistribution matrix as shown in Figure 6.1. In Step 2, notice we use the mode location of $R_i^b[n,]$ as normal density function center instead of the mean. In reality, the shape of the row vector, $R_i^b[n,]$ is often a bit left-skewed and not perfectly normal. After conducting multiple tests, we found that using the

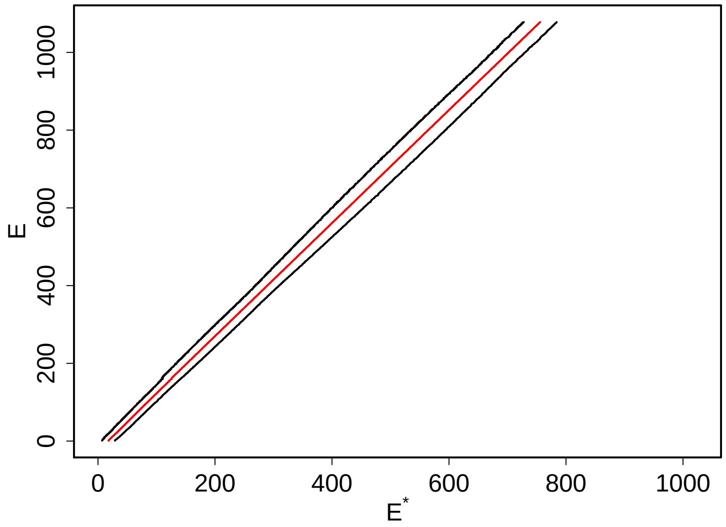

Figure 6.4: Boundaries used to separate R into R^b and R^s. Two black lines indicate the boundaries of the redistribution matrix elements that we will consider uncertainty of. We build the matrix R^b by keeping the elements between these two lines, and forcing other elements to be zero. And R^s can be simply constructed by $R - R^b$. The red line indicates the location of elements containing largest value for each row of R_0

mode location of $R_i^b[n,]$ as normal density function center works better than using mean, for the approximation of the large value elements, where we put high priority. In Step 4, we found that using the one, three and ten largest principle components count for 83.3%, 92.1%, and 94% of the variance among $\ominus$, respectively. Since the additional variance explained by using $J = 10$ rather than $J = 3$ is small, we use $J = 3$ in the following numerical analysis. Moreover, smaller J means easier implementation for both pragmatic and fully Bayesian methods.

6.1.4 Sampling Redistribution Matrix

Having summarized the redistribution matrix, we can build up a prior distribution for it, $\pi(R)$. It would be similar to, but slightly more complicated than, setting an effective area prior. In this section, we detail the steps for sampling a redistribution matrix R_{new} and evaluating the prior probability of R_{new}.

Sampling the Redistribution Matrix:

Step 1: Simulate $e_j \sim \mathcal{N}(0,1)$ independently for $j = 1, ..., J$.

Set $e = \{e_1, ..., e_J\}$ and $\Psi_{new}(e) = \bar{\Psi} + \sum_{j=1}^{J} e_j r_j v_j$

Step 2: Once the $\Psi_{new}(e)$ is simulated, R^b_{new} can be constructed, as follows.

Set $\mu_{new;n} = \Psi_{new}[2n-1]$, and $\sigma^2_{new;n} = \Psi_{new}[2n]$, and

$$R^b_{new}[n,k] = \frac{1}{\sqrt{2\pi\sigma^2_{new;n}}} exp\{-\frac{(k-\mu_{new;n})^2}{2\sigma^2_{new;n}}\}$$

Notice this step is only for the elements of R^b, which are the entires between the two black lines in Figure 6.4

Step 3: Set the new redistribution matrix $R_{new} = (R^*_0 - R_0) + R^b_{new} + \bar{R}^s$,

where R^*_0 is the user-generated observation-specific redistribution matrix, R_0 is the default redistribution matrix of the library $\mathcal{R}$, and $\bar{R}^s$ is the mean of the redistribution matrix library for the small elements, which are the entires outside two black lines in Figure 6.4.

Step 4: Modify R_{new} to ensure that it is a valid redistribution matrix: negative entries, if any are set to zero, and each entry is divided by the row sum to make the row sum to one.

In Step 1, all the values of the new simulated Ψ_{new} need to be positive to be a valid parameter vector. In practice, it is rare to have negative values, because $\ominus$ contains all positive values, and the PCA typically results in an all-positive Ψ_{new}, too. In Step 3, the term $R^*_0 - R_0$ is the adjustment for the difference between observation-specific redistribution matrix and the redistribution matrix library. Adding $\bar{R}^s$ directly into R_{new} means that the uncertainty among the small value elements is ignored.

Overall, this algorithm allows us to sample a new distribution matrix only by sampling independent standard normal deviations e and constructing the matrix from e. We will use the notation $R(e)$ to represent the construction of R from e. Thus the prior for R obtained from the prior for e:

$$\pi(R) = \prod_{j=1}^{J} \pi(e_j) \tag{6.1}$$

Since e is simulated independently from $\mathcal{N}(0, 1)$, the calculation of the prior for $\pi(R)$ becomes straightforward. We expect the effect of probability transformation in Step 4 is minimal.

Figure 6.5 and 6.6 illustrate the performance of this redistribution matrix sampling algorithm, compared to redistribution matrix library. 50 matrix elements are picked for the comparison, 25 elements containing large values (Figure 6.5) and the other 25 containing relatively small values (Figure 6.6). The histogram demonstrates the probability density of certain matrix elements from 1000 redistribution matrix samples of the library, while the red curve demonstrates the probability density of the same matrix element from 1000 simulated redistribution matrices using this sampling algorithm. These two figures show that in general the redistribution matrix sampling algorithm can capture the structure the uncertainty of the redistribution matrix. It is not working perfectly when the row index is below 80, corresponding to low energy bins below below 1 Kev.

However, note that using this sampling algorithm, we succeed to reduce the dimension of the redistribution matrix from one million to three, which favors the implementation of the pragmatic Bayesian and fully Bayesian methods. In the following sections of this chapter, we will discuss how this sampling algorithm is used in the pragmatic Bayesian and fully Bayesian methods for calibration uncertainty in the photon redistribution matrix.

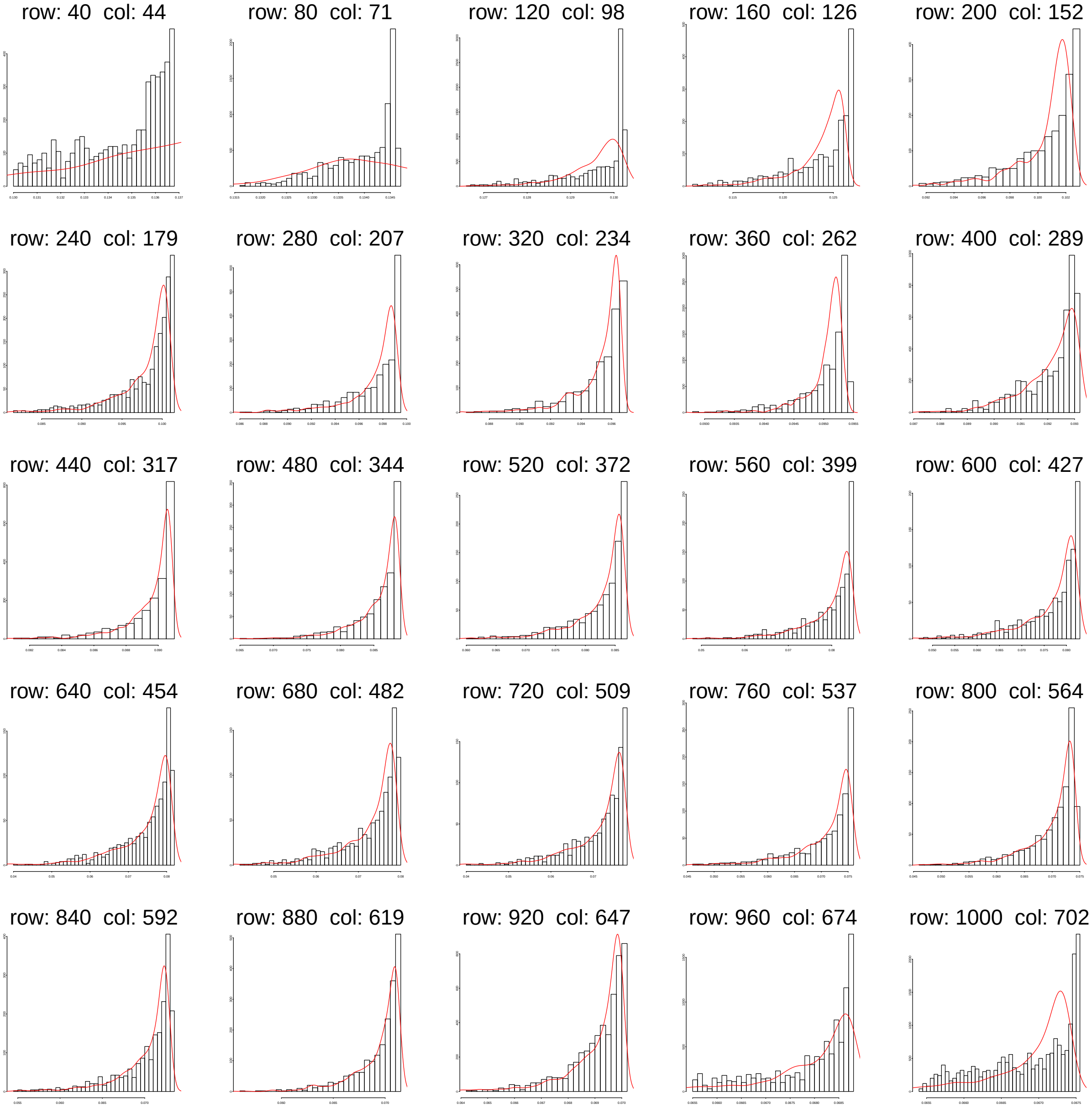

Figure 6.5: Redistribution matrix sampling algorithm for 25 large value elements. The histogram demonstrates the probability density of certain matrix elements from 1000 redistribution matrix samples of the library, while the red curve demonstrates the the probability density of the same matrix element from 1000 simulated redistribution matrices using this sampling algorithm.

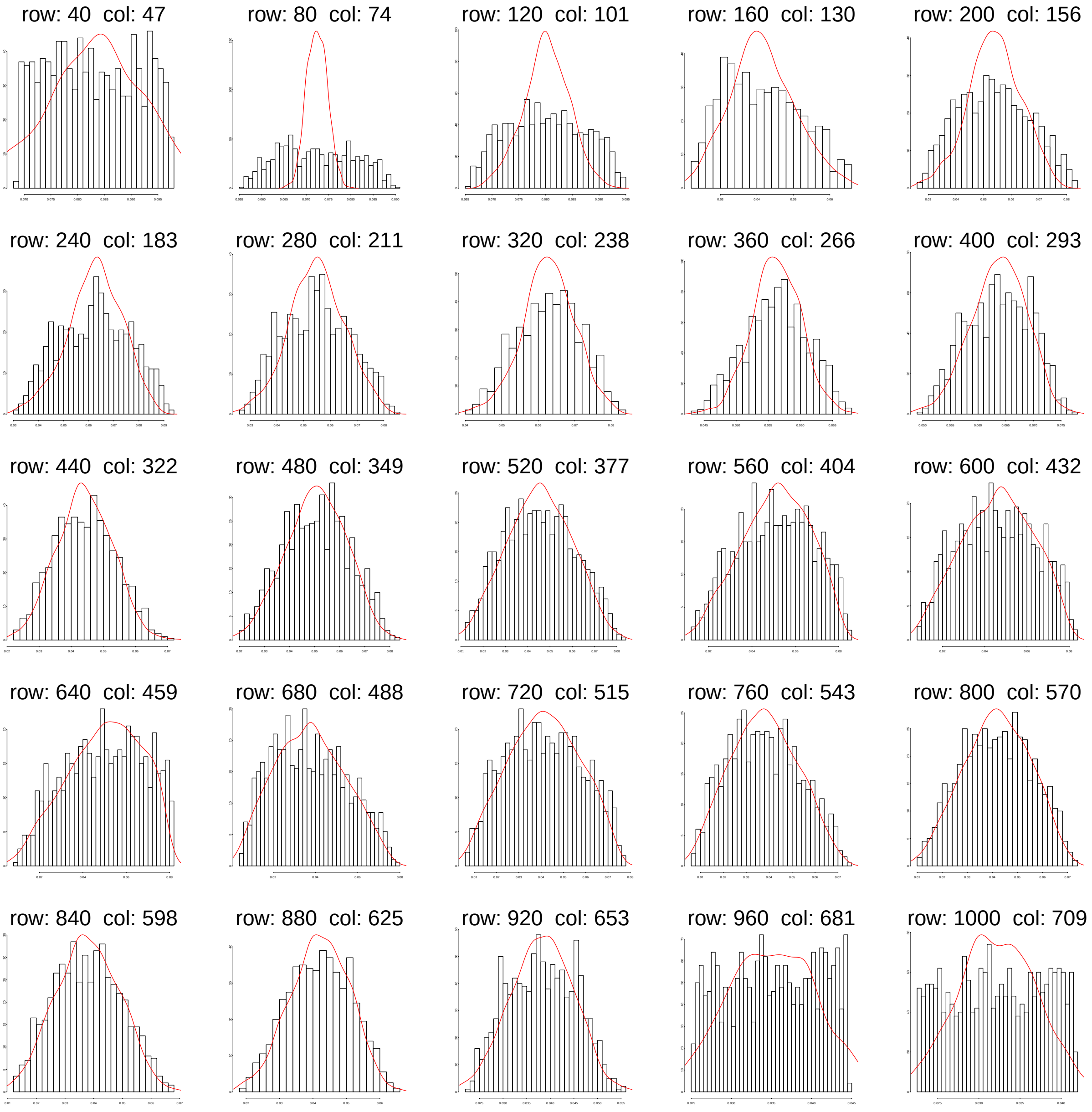

Figure 6.6: Redistribution matrix sampling algorithm for 25 relatively small value elements. The histogram demonstrates the probability density of certain matrix elements from 1000 redistribution matrix sample of the library, while the red curve demonstrates the the probability density of the same matrix element from 1000 simulated redistribution matrices using this sampling algorithm.

6.2 Pragmatic Bayesian Method

For the following sections of this chapter, we will only focus on calibration uncertainty of photon redistribution matrix, while the effective area curve will be fixed to be the default one. First, we illustrate how the choice of the redistribution matrix affects the spectral analysis result and why incorporating the uncertainty of the redistribution matrix is necessary. Secondly, the pragmatic Bayesian method for photon redistribution matrix is described.

6.2.1 The Effect of the Choice of Photon Redistribution Matrix

Consider the spectral model, Equation (2.1), and suppose we adopt the Bayesian framework, where the spectral parameter θ and the photon redistribution matrix R are treated as unknown quantities, $\psi = (\theta, R)$, and the effective area curve A becomes a part of known information, $\mathcal{I}$.

Assuming the prior distribution for θ and R are independent, we have the posterior distribution as

$$p(\theta, R|Y) \propto L(Y|\theta, R)\pi(\theta)\pi(R), \tag{6.2}$$

First, as in Section 2.2 which handles the effective area curve, we also have the standard method for R, which assumes a fixed and known $R = R_0^*$. R_0^* is the nominal redistribution matrix associated with this observation. Thus, this standard method estimates the posterior distribution of θ, using

$$p_{\text{std}}(\theta|Y, R_0^*) \propto L(Y|\theta, R_0^*)\pi(\theta), \tag{6.3}$$

However, in reality, R_0^* is usually different from the true unknown value of R. Pretending

$R = R_0^*$ can lead to misleading estimates of θ and significantly underestimate the error bars associated with these estimates.

To illustrate how the choice of the redistribution matrix affects the spectral analysis result, we conduct a simulation study. In Section 6.4, more simulation studies will be conducted. Here we give detailed results under the simulation setting called Simulation II in Section 6.4. In particular, we simulate an absorbed power-law source model with three parameters (power-law index Γ, absorption column density N_{H}, and normalization) using the `fake_pha` routine in Sherpa. The data set was simulated without background contamination using the XSPEC model `wabs*powerlaw` and a default (ARF) for ACIS-S. Here we give detailed results for Simulation II which sets $\Gamma = 1$, $N_{\mathrm{H}} = 10^{21}\mathrm{cm}^{-2}$, with 10^5 counts, and uses an extremal redistribution matrix from the redistribution matrix library $\mathcal{R}$. We label it R_{ext}.

Figure 6.7 shows different standard method results of Simulation II, corresponding to fitting the model using 15 different redistribution matrices. It demonstrates that the choice of redistribution matrix can highly influence the spectral analysis result. If we use the wrong redistribution matrix (in practice, we mostly use R_0^*), the inference of spectral parameters could be biased, and the confidence interval may undercover. Only when specifying the redistribution matrix as the true one, R_{ext}, can we draw a valid conclusion from the standard method.

Notice that the data set in Simulation II is relatively large, containing 10^5 photon counts. The bigger the data set, the more information there is, and the narrower the error bars for the spectral parameters are. On the other hand, specifying the wrong redistribution matrix for large data sets could lead to obvious bias and anti-conservative confidence intervals.

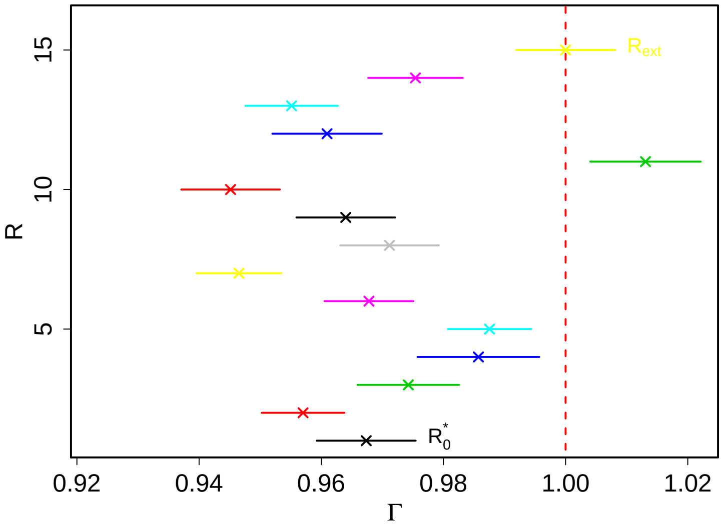

Figure 6.7: Effect of the choice of 15 different redistribution matrices. 15 different redistribution matrices including the default one R_0^* and the extremal true one R_{ext} are selected to conduct the standard method for Simulation II. For each choice of the redistribution matrix, the star represents the posterior mean of the spectral parameter Γ, and the horizontal bar represents the one-standard-deviation confidence interval. The vertical broken red line is the true value of Γ.

6.2.2 Pragmatic Bayesian Algorithm

Same as we did for the uncertainty of the effective area curve in Section 2.4 and 4.2, we now propose the pragmatic Bayesian method for uncertainty of redistribution matrix and the corresponding sampling algorithm. With the "pragmatic assumption" $p(R|Y) = \pi(R)$, we have the posterior distribution for the pragmatic Bayesian method:

$$p_{\text{pB}}(\theta, R|Y) = p(\theta|R, Y)\pi(R) \tag{6.4}$$

Because the pragmatic Bayesian method essentially assumes that the observed data carry no information about the redistribution matrix, and it will not narrow the uncertainty in the choice of redistribution matrix. The pragmatic Bayesian method accounts for the calibration uncertainty in a conservative manner.

With the prior for redistribution matrix set up in Section 6.1, the implementation of the pragmatic Bayesian method for redistribution matrix is similar to that for the effective area curve. The only difference is the way we sample the redistribution matrix. Once the redistribution matrix R is simulated, we can still use `PyBLoCXS` to sample spectral parameters θ from $p(\theta|Y, R)$.

Because of the complexity of sampling the redistribution matrix, the CPU time for the first step of the pragmatic Bayesian is around 8 seconds in the simulation of Section 6.2, which costs even more than the Sherpa fit after each redistribution matrix update. Thus MH within PCG Sampler (Section 4.1) for redistribution matrix pragmatic Bayesian is even more expensive. (A run of 3000 MCMC iterations requires 13-15 hours.) For later analysis of the pragmatic Bayesian method, we will only consider the efficient Iterated MH within PCG Sampler (Section 4.2), since it fully leverages each update of the redistribution matrix.

Iterated Pragmatic Bayesian Sampler:

For $t = 0, 1, ..., T$

Step 1: Simulate $e_j^{(tM+1)} \sim \mathcal{N}(0,1)$ for $j = 1, ..., J$ and set $e_{\text{pB}}^{(tM+1)} = (e_1^{(tM+1)}, ..., e_J^{(tM+1)})$ and $R_{\text{pB}}^{(tM+1)} = R(e_{\text{pB}}^{(tM+1)})$.

Step 2: For $i = 1, ..., I$, simulate $\theta^{(Mt+i/I)} \sim \mathcal{K}_{\text{pyB}}(\theta|\theta^{(Mt+(i-1)/I)}; Y, R^{(t+1)})$.

Step 3: For $i = 2, ..., M$, simulate $\theta_{\text{pB}}^{(Mt+i)} \sim \mathcal{K}_{\text{pyB}}(\theta|\theta^{(Mt+i-1)}; Y, R^{(t+1)})$

The details of the algorithm can be found in Section 4.1 where we also discuss the procedure to obtain the optimal number M.

6.3 Fully Bayesian Method

The fully Bayesian Method, on the other hand, allows the data to provide information about the redistribution matrix and eventually narrow the uncertainty in the choice of redistribution matrix. In other words, the data themselves carry information not only for θ, but also for R. This is a principled Bayesian procedure. We have the posterior distribution of the spectral parameters θ and the redistribution matrix R given observed data Y:

$$p_{\text{fB}}(\theta, R|Y) \propto L(Y|\theta, R)\pi(\theta)\pi(R), \tag{6.5}$$

We will implement fully Bayesian method with the same procedure called Pragmatic Proposal Sampler as described in Section 4.3.2. To do this we run the Iterated Sampler described in Section 6.2.2 to obtain $\{e_{\text{pB}}^{(tM+1)}, \theta_{\text{pB}}^{(tM+1)}, \ldots, \theta_{\text{pB}}^{(tM+M)}$, for $t = 0, \ldots, T\}$ and regress the M replicates of θ, (i.e., $\theta_{\text{pB}}^{(tM+1)}, \ldots, \theta_{\text{pB}}^{(tM+M)}$) on each $e_{\text{pB}}^{(tM+1)}$ using multivariate Gaussian regression, as in Section 4.3.3. This results in a multivariate normal approximation to $p_{\text{pB}}(\theta \mid R, Y)$ that we combine with a multivariate standard normal distribution for e to form $Q(\theta, e)$. With this proposal Q, we implement the MH algorithm for the fully Bayesian method, as in Section 4.3.4.

In particular, this choice of Q serves as the needed over-dispersed approximation to $p_{\text{pB}}(\theta, R \mid Y)$. In this context we run the Iterated MH within PCG Sampler to obtain the approximation, Q, because Q does not need to exactly match $p_{\text{pB}}(\theta, R \mid Y)$.

Thus we have:

Pragmatic Proposal Sampler:

For $t = 1, 2, ..., T$

Step 1: Randomly sample u_1 from a standard uniform distribution.

If $u_1 < p_m$, go to Step 2, Partial Random Walk Sampler

else, go to Step 3, Independence Sampler

Step 2: For $j = 1, ..., J$, Sample $e_j^{\text{prop}} \sim \mathcal{N}(\mu = e_j^{(t)}, sd = \sigma_e)$

Set $e^{\text{prop}} = (e_1^{\text{prop}}, ..., e_J^{\text{prop}})$ and $R^{\text{prop}} = R(e^{\text{prop}})$.

Sample $\theta^{\text{prop}} \sim Q_{\text{MVN}}(\theta | e^{\text{prop}})$

$$\alpha = \frac{p(R^{\text{prop}}, \theta^{\text{prop}} | Y) Q_{\text{MVN}}(\theta^{(t)} | e^{(t)})}{p(R^{(t)}, \theta^{(t)} | Y) Q_{\text{MVN}}(\theta^{\text{prop}} | e^{\text{prop}})},$$

go to Step 4

Step 3: For $j = 1, ..., J$, Sample $e_j^{\text{prop}} \sim \mathcal{N}(0, 1)$

Set $e^{\text{prop}} = (e_1^{\text{prop}}, ..., e_J^{\text{prop}})$ and $R^{\text{prop}} = R(e^{\text{prop}})$.

Sample $\theta^{\text{prop}} \sim Q_{\text{MVN}}(\theta | e^{\text{prop}})$

$$\alpha = \frac{p(R^{\text{prop}}, \theta^{\text{prop}} | Y) Q_{\text{MVN}}(\theta^{(t)}, e^{(t)})}{p(R^{(t)}, \theta^{(t)} | Y) Q_{\text{MVN}}(\theta^{\text{prop}}, e^{\text{prop}})},$$

go to Step 4

Step 4: Randomly sample u_2 from a standard uniform distribution.

$$(e^{(t+1)}, R^{(t+1)}, \theta^{(t+1)}) = \begin{cases} (e^{\text{prop}}, R^{\text{prop}}, \theta^{\text{prop}}) & \text{if } u_2 < \alpha \\ (e^{(t)}, R^{(t)}, \theta^{(t)}) & \text{otherwise} \end{cases}.$$

More discussion about this algorithm can be found in Section 4.3.4. Here, we emphasize that this algorithm can also work for the redistribution matrix.

Table 6.1: The eight simulations used to compare the standard, pragmatic Bayesian, and fully Bayesian methods.

	Redistribution Matrix		Nominal Counts		Spectal Model	
	Default[⋆]	Extreme[⋆⋆]	10^5	10^4	Hard[†]	Soft[‡]
SIMULATION I		X	X		X	
SIMULATION II		X	X			X
SIMULATION III		X		X	X	
SIMULATION IV		X		X		X
SIMULATION V	X		X		X	
SIMULATION VI	X		X			X
SIMULATION VII	X			X	X	
SIMULATION VIII	X			X		X

[⋆] The default effective area curve, R_0, from the redistribution matrix calibration library

[⋆⋆] An extreme effective area curve , R_{ext}, from the redistribution matrix calibration library

[†] An absorbed powerlaw with $\Gamma = 2$, $N_{\text{H}} = 10^{23}/\text{cm}^2$

[‡] An absorbed powerlaw with $\Gamma = 1$, $N_{\text{H}} = 10^{21}/\text{cm}^2$

6.4 Simulation Studies

To compare the performance of the standard, pragmatic Bayesian and fully Bayesian methods, eight data sets were simulated from an absorbed power-law source with three parameters (power-law index Γ, absorption column density N_H, and normalization) using the **fake_pha** routine in Sherpa. This simulation setting is exactly the same as we did for the effective area curve, in Section 5.1, except that, we use the default effective area curve throughout all eight simulations and use the extremal redistribution matrix R_{ext} for the first four simulations and the default redistribution matrix R_0 for the last four simulations. The simulation setting is shown in detail, in Table 6.1

We make the model comparison between the standard, the pragmatic Bayesian and the fully Bayesian methods by fitting all the eight data sets with the default redistribution matrix R_0. This means that first four data sets are fit with the wrong redistribution matrix, while the other four are using the right redistribution matrix. Figures 6.8 and 6.9 demonstrates the posterior distributions, intervals and fitted values for Γ computed with each of three methods run on each simulation. The standard, pragmatic Bayes and Fully Bayesian methods are represented by black, blue, and red curves and intervals, respectively. The posterior means (fitted values) are represented by a "×" and their 68.2% error bars are represented by horizontal bars.

From Figure 6.8 and 6.9, we observe almost identical results as in Section 5.1.

- The standard method that does not account for calibration uncertainty exhibits the narrowest error bars, the pragmatic Bayesian method accounting for the uncertainty conservatively has the widest error bars, and those of the fully Bayesian method are in between.
- When the correct R can be specified, the standard method should be used; see Simulations V–VIII.
- Simulations I–IV illustrates the most common scenarios, when the true R is unknown. In these cases, both the standard and pragmatic Bayesian methods exhibit more bias than the fully Bayesian method.
- Photon counts (data size) are also a crucial factor that influences the result of the fully Bayesian method. With more information in the data, the fully Bayesian method can perform better in selecting more consistent redistribution matrices to the data, thus helping the fitted spectral parameters to shift toward the true values; see Simulations I and II.

6.5 Computational Time

In this section, we discuss the computational time of the the pragmatic Bayesian and fully Bayesian methods for uncertainty of the redistribution matrix. Due to the complicated

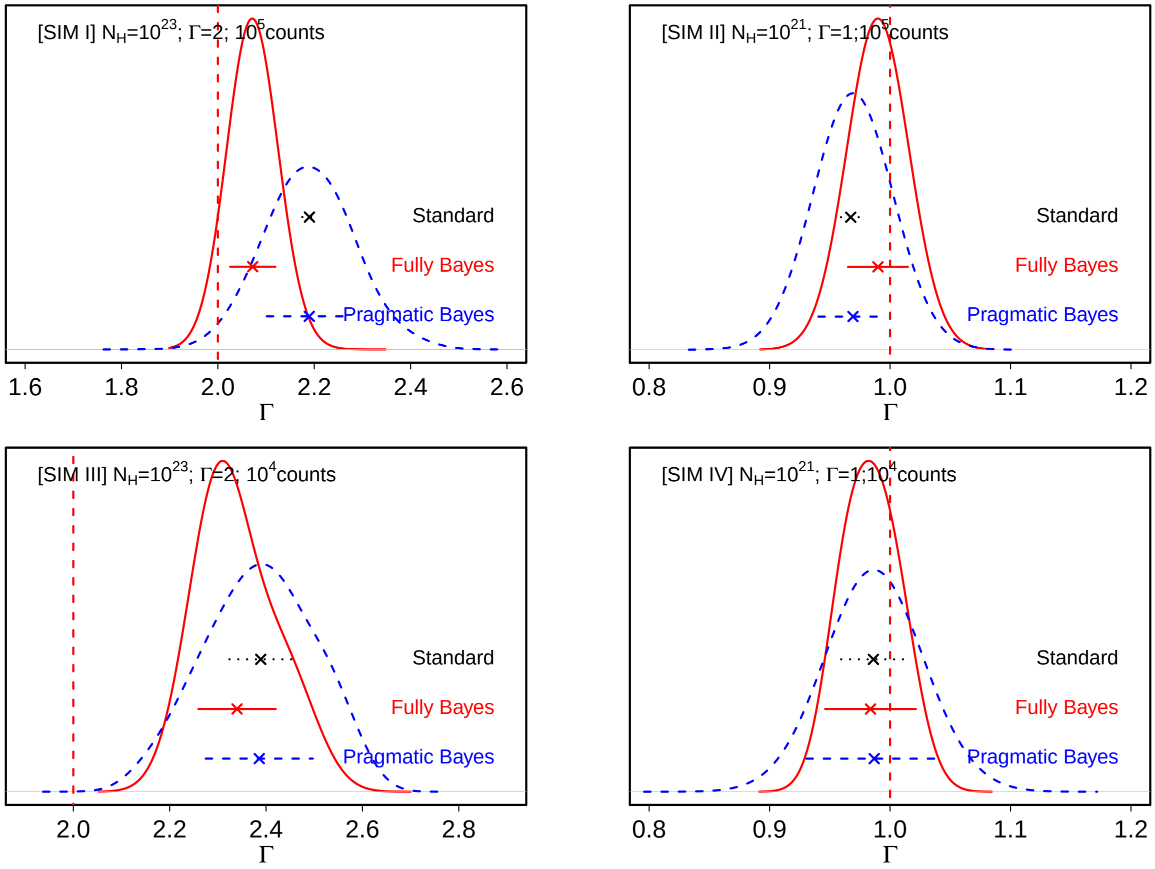

Figure 6.8: Results for Simulations I–IV for redistribution matrix. The panels show the posterior distributions (curves), fitted values (×), and 68.2% error bars (horizontal bars) for Γ. Results for the standard, pragmatic Bayesian, and fully Bayesian methods are plotted in black, blue, and red, respectively. The true value of Γ is given by the red broken vertical lines. These simulations consider the situation in which the default redistribution matrix is misspecified to a degree that is consistent with the variability of the the calibration library. Because the standard method uses this misspecified matrix, it preforms poorly. Both the pragmatic and the fully Bayesian methods avoid assuming that R is known without error allowing them to preform better. Of these two, the fully Bayesian method provides both estimates of Γ that are closer to its true values and narrower error bars, especially in Simulation I and II.

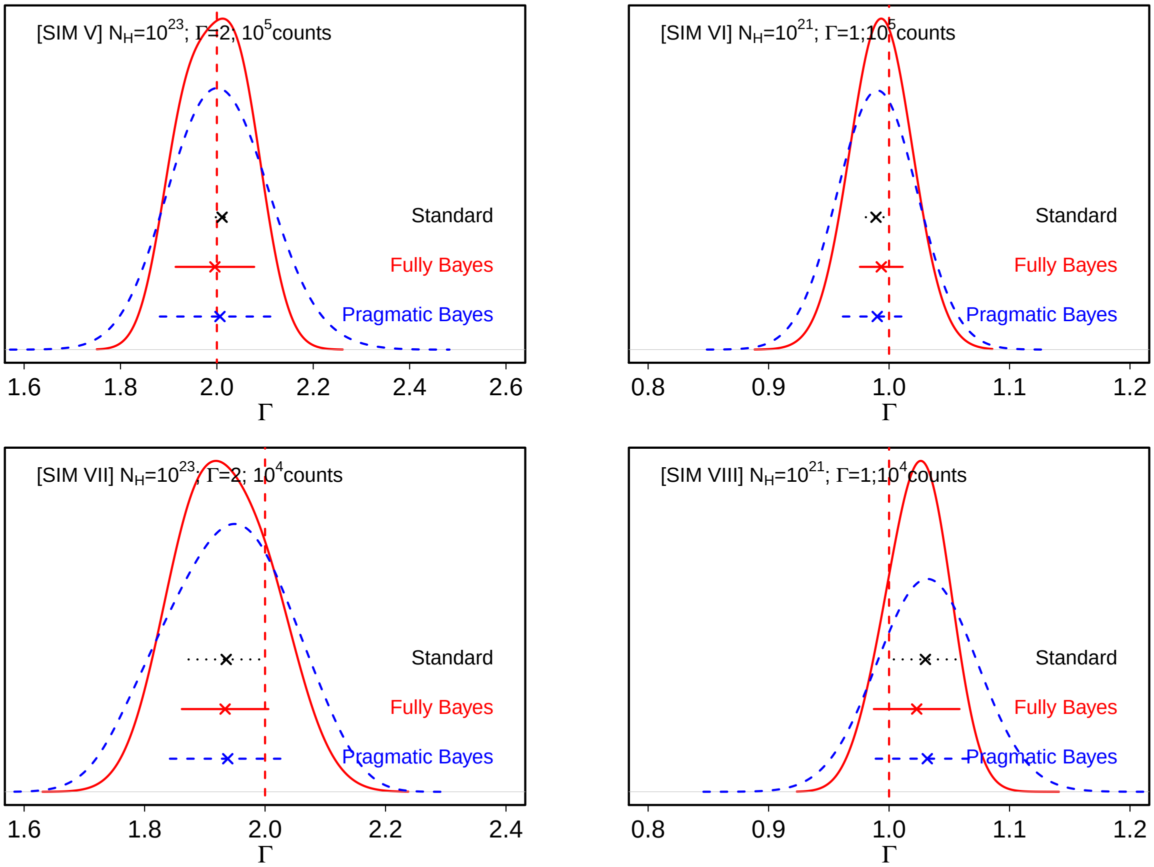

Figure 6.9: Results of Simulations V–VIII for redistribution matrix. Since we assume that the true redistribution matrix is known in these four simulations, as expected, the standard method performs the best.

structure of the redistribution matrix, the pragmatic Bayesian and fully Bayesian methods cost much more time than those for the uncertainty of the effective area curve. Hence computational time becomes an issue, should we use these methods in practice.

Table 6.2 exhibits the computational time for all the samplers involved in this thesis. For effective area curve methods, Simulation II from Section 5.1 is selected as the example, and for redistribution matrix methods, Simulation II from Section 6.4 is selected. Both examples have the same setting for spectral parameters. The computational time in Table 6.2 represents the cost for 3000 iterations of these samplers.

We focus on the combination of iterated pragmatic Bayesian and the pragmatic proposal sampler for fully Bayesian solution (that is, Column 4, 6 for effective area curve, and Column 9, 10 for redistribution matrix in Table 6.2). For effective area curve methods, the main computational time is due to setting up the proposal density after each update of the effective area curve (this step involves the sherpa fit). Iterated pragmatic Bayesian highly leverage each update of effective area curve and makes multiple use of the same proposal density, thus reduces the computational time (from 6 hours to 40 minutes in Table 6.2). The pragmatic proposal sampler for fully Bayesian, on the other hand, takes the samples of iterated pragmatic Bayesian as the proposal, and thus requires little additional computational time (15 minutes in Table 6.2). For the redistribution matrix methods, the computational time has two main sources, (1) the sample of new redistribution matrix and setting it up in a usable format, due to the complicated sampling algorithm in Section 6.1.4 and (2) the proposal setting after each update of redistribution matrix for the pragmatic Bayesian. Iterated pragmatic Bayesian can still highly leverage each update of redistribution matrix, reducing two sources of computational time simultaneously (from 14 hours to 1.5 hours, in Table 6.2). However, the pragmatic proposal sampler for fully Bayesian requires to sample a new matrix and set it up in a usable formate in every iteration, and therefore, the first source of computational time becomes the main issue. In Table 6.2, we can see fully Bayesian for the redistribution matrix costs 8 hours, compared to 15 minutes of the fully Bayesian for effective area curve.

Table 6.2: Computational Time Summary

	Effective Area Uncertainty					Redistribution Matrix Uncertainty			
Data Set	Simulation II from Section 5.1					Simulation II from Section 6.4			
Methods	Standard	Pragmatic Bayesian		Fully Bayesian		Standard	Pragmatic Bayesian		Fully Bayesian
Samplers	`PyBLoCXS`	Original PB	Iterated PB	Gibbs	Prag Proposal	`PyBLoCXS`	Original PB[*]	Iterated PB	Prag Proposal
Tuning Setting[†]			$M = 10$	$p_m = 0.5$				$M = 10$	
Sampler Section	`PyBLoCXS`	4.1	4.2	4.3.1	4.3.4	`PyBLoCXS`		6.2.2	6.3
Time[‡]	10min	6hours	40min	3hours	15min[**]	10min	14hours	1.5hours	8hours[**]

[*] This algorithm similar to the case of effective areas in Section 4.1. We omit the details of implementation in the thesis.

[**] The computational time for the pragmatic proposal samplers does not include the time to draw the samples of the pragmatic Bayesian.

[†] Only the tuning parameters that affect computational time are included

[‡] The computational time can vary due to the computing machines we use.

Computational Time Summary. We focus on the combination of iterated pragmatic Bayesian and the pragmatic proposal sampler for fully Bayesian solution (that is, Column 4, 6 for effective area curve, and Column 9, 10 for redistribution matrix.) Iterated pragmatic Bayesian, leveraging each update of effective area curve or redistribution matrix making multiple use of the same proposal density, reduced the computational time significantly. The pragmatic proposal samplers use the samples from iterated pragmatic Bayesian as proposal density, and requires little additional time for effective area uncertainty. However, due to the complicated structure of redistribution matrix, each step of setting up for a new redistribution matrix for the fully Bayesian method costs a large amount of time. Hence, the pragmatic proposal distribution for redistribution matrix uncertainty still requires 8 hours.

Chapter 7

Discussion

In this thesis we demonstrate the advantage of a fully Bayesian method that accounts for calibration uncertainty. Compared to the pragmatic Bayesian method of Lee *et al.* (2011) the fully Bayesian method gives estimates with smaller bias and narrower error bars. As with the pragmatic method, the fully Bayesian method requires large-counts data sets to deliver significant gains over the standard fixed effective area method. In low-counts data sets, uncertainty stemming from random fluctuations in the counts swamps that due to calibration. The advantage of the fully Bayesian method stems from its use of a calibration library to transform systematic errors stemming from calibration uncertainty into statistical errors. In this way it accounts for calibration uncertainty by shifting the fitted values of spectral parameters. From a scientific perspective, this is preferable to increasing their error bars, the mechanism by which the pragmatic Bayesian method accounts for calibration uncertainty. Fitting a spectral model under the fully Bayesian method poses significant computational challenges. We illustrate how we solve this problem by leveraging the pragmatic Bayesian fit to deliver parameter values simulated under this fully Bayesian posterior distribution. This strategy allows us to simultaneously fit the calibration product and the spectral parameters. Thus, we are able to use information in large-count observed spectra to narrow the uncertainty for the calibration product.

We focus on the application of the fully Bayesian methods to account for uncertainty in the effective area curve in X-ray spectral analysis from Chapter 2 to 5. In Section 5.1 and Section 5.2, the simulation study is conducted to prove that fully Bayesian is able to fit the effective area curve and the spectral parameters simultaneously. When the true effective area curve is unknown, the fully Bayesian can reduce the bias of fits of the spectral parameters, and thus is recommended to apply in reality.

The general techniques we employ, however, have broad applicability in handling systematic errors. Obvious extensions include accounting for uncertainty in other calibration products, such as photon redistribution matrices, exposure maps, and point spread functions. Because all of these calibration products exhibit more complex structure than an effective area curve—they are represented by matrices rather than vectors—more sophisticated methods will be needed to summarize their calibration libraries into concise and useable forms. In Chapter 6, we develop a method to represent the photon redistribution matrix, effectively reducing the dimension from 10^6 to 3. This algorithm enables us to conduct the same fully Bayesian method to account for the uncertainty of the photon redistribution matrix. The simulations in Section 6.4 demonstrates that the advantages of fully Bayesian method persist when counting different type of calibration product. However, the computational cost is dramatically increased due to the complexity of the redistribution matrix. As future work, we need to develop more efficient algorithms and apply the fully Bayesian to the real data.

Another challenge is to incorporate calibration uncertainty of effective area curve and photon redistribution matrix simultaneously into the spectral analysis, that is, we aim to make inference for the posterior distribution $p(\theta, A, R|Y)$, where there is uncertainty in both effective area curve and photon redistribution matrix. As before, we consider a fully Bayesian solution.

With the use of the samplers we developed in the thesis, we propose a collapsed Gibbs fully Bayesian sampler:

Collapsed Gibbs Fully Bayesian Sampler :

For $t = 0, 1, 2, \ldots, T$,

Step 1: Sample $(A^{(t+1)}, \theta^{(t+1/2)}) \sim p(A, \theta | R^{(t)}, Y)$

Step 2: Sample $(R^{(t+1)}, \theta^{(t+1)}) \sim p(R, \theta | A^{(t+1)}, Y)$

We do not keep the intermediate iterations, $\theta^{(t+1/2)}$, thus it is called collapsed Gibbs sampler. In theory, this sampler can be conducted now by combining the samplers we have developed in the thesis. Step 1 involves the fully Bayesian for effective area in Section 4.3.4, and Step 2 involves the fully Bayesian for redistribution matrix in Section 6.3. If we succeeded to implement this sampler, the correlations between calibration products and spectral parameters can be analyzed through the Monte Carlo draws. However, we expect heavy computational burden of this method, especially if at each step, we need to use the pragmatic Bayesian to build a new proposal density. Clearly, more sophisticated method is required to overcome this problem.

Another approach to account for calibration uncertainty of A and R simultaneously is to build up a new calibration product combining A and R. Specifically, Equation (2.1) for the photon detecting model can be written in the following matrix format:

$$Y \sim Pois\left(R^{\top} \dot{A} \Lambda + B\right), \tag{7.1}$$

where Y is the observed photon count vector with dimension $K \times 1$, R is the photon redistribution matrix with dimension $N \times K$, $\dot{A}$ is an $N \times N$ diagonal matrix constructed from the effective area curve with $\dot{A}[n;n] = A[n]$, Λ is the spectral intensity vector with dimension $N \times 1$ and B is the background intensity vector with dimension $K \times 1$.

In Equation (7.1), we can treat $\Omega = R^\top \dot{A}$ as a new calibration product, which encodes the information from both A and R. Suppose we have the paired calibration samples $\{(A_1, R_1), ..., (A_L, R_L)\}$, and the calibration library for Ω can be constructed as $\{\Omega_1 = R_1^\top \dot{A}_1, ..., \Omega_L = R_L^\top \dot{A}_L\}$. We expect the algorithm to summarize Ω would be similar to the one we developed for the photon redistribution matrix in Section 6.1.3, since Ω and R have similar matrix formats. Once the prior for Ω is set up, the fully Bayesian method can be implemented following the same procedure as in the case of the effective area curve. This approach seems promising and worthy of further exploration for accounting for calibration uncertainty of A and R simultaneously. It is more straightforward than the collapsed Gibbs approach.

Besides calibration uncertainty, our methods can be used to account for other sources of systematic errors, such as uncertainty in a background model. In our fit of the radio-loud quasars in Section 5.3 and background-structured isolated neutron star RXJ1856 in Section 5.4, for example, we used a highly-structured background model that was originally fit to the blank-sky data provided by the *Chandra* X-ray Center. Only the normalization of this background model was fit to the individual quasars. Just like an effective area curve, this background model is a vector that can only be specified with uncertainty. Ignoring this uncertainty can lead to biases and systematic errors. Similarly, the comprehensive atomic line emissivity database, AtomDB (Foster *et al.* (2012)) is often used to specify a spectral models for X-ray data. While this database has been compiled by carefully combining empirical observations with theoretical calculations, its entries are not known exactly. Like a calibration product its errors exhibit complex high-dimensional correlations that cannot be summarized with simple error bars for each entry. A better strategy is to compile an "atomDB library" akin to a calibration library that can then be modeled to fully integrate uncertainty in atomDB into individual spectral analyses. In principle, we can image fitting models that simultaneously account for uncertainly both in multiple calibration products and in multiple model components. In practice, this will involve significant modeling challenges such as accounting for correlations between calibration products and/or model components. Sophisticated computation methods will also be required, and large data sets needed to narrow uncertainty on multiple sources of systematic error. Although such work will involve substantial effort, it is likely to pay significant dividends in reducing bias stemming from calibration and model misspecified, at least when large-count data sets are available.

Bibliography

Anders, E. and Grevesse, N. (1989). Abundances of the elements - meteoritic and solar. *Geochimica et Cosmochimica Acta* **53**, 197–214.

Anderson, T. W. (2003). An introduction to multivariate statistical analysis, 3rd edition .

Bevington, P. and Robinson, D. (1992). Testing the fit. *Data Reduction and Error Analysis for the Physical Sciences* 194–198.

Bishop, C. M. *et al.* (2006). *Pattern recognition and machine learning*, vol. 1. springer New York.

Bridle, S., Crittenden, R., Melchiorri, A., Hobson, M., Kneissl, R., and Lasenby, A. (2002). Analytic marginalization over cmb calibration and beam uncertainty. *Monthly Notices of the Royal Astronomical Society* **335**, 4, 1193–1200.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of Markov Chain Monte Carlo*. CRC Press.

Brooks, S. P. and Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics* **7**, 4, 434–455.

Dieck, R. H. (1992). *Measurement uncertainty: methods and applications*. Instrument Society of America Research Triangle Park, NC.

Drake, J. J., Marshall, H. L., Dreizler, S., Freeman, P. E., Fruscione, A., Juda, M., Kashyap, V., Nicastro, F., Pease, D. O., Wargelin, B. J., and Werner, K. (2002). Is rx j1856.5-3754 a quark star? *The Astrophysical Journal* **572**, 996–1001.

Drake, J. J., Ratzlaff, P., Kashyap, V., Edgar, R., Izem, R., Jerius, D., Siemiginowska, A., and Vikhlinin, A. (2006). Monte carlo processes for including chandra instrument response uncertainties in parameter estimation studies. In *Astronomical Telescopes and Instrumentation*, 62701I–62701I. International Society for Optics and Photonics.

Foster, A., Ji, L., Smith, R., and Brickhouse, N. (2012). Updated atomic data and calculations for x-ray spectroscopy. *The Astrophysical Journal* **756**, 2, 128.

Gelfand, A. E. and Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association* **85**, 410, 398–409.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science* 457–472.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , 6, 721–741.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika* **57**, 1, 97–109.

Heinrich, J. and Lyons, L. (2007). Systematic errors. *Annual Review of Nuclear and Particle Science* **57**, 1, 145.

Herve, A., Rauw, G., and Naze, Y. (2013). Hot gas distribution in the wind of ζ pup and ζ ori. In *Massive Stars: From α to Ω, held 10-14 June 2013.*

Ho, W. C., Kaplan, D. L., Chang, P., van Adelsberg, M., and Potekhin, Alexander, Y. (2007). Magnetic hydrogen atmosphere models and the neutron star rx j1856.5-3754. *Monthly Notices of the Royal Astronomical Society* **375**, 821–830.

Jolliffe, I. (2005). *Principal component analysis.* Wiley Online Library.

Kelly, B. C. (2007). Some aspects of measurement error in linear regression of astronomical data. *The Astrophysical Journal* **665**, 2, 1489.

Kutner, M. L. and Ulich, B. (1981). Recommendations for calibration of millimeter-wavelength spectral line data. *The Astrophysical Journal* **250**, 341–348.

Lee, H., Kashyap, V. L., van Dyk, D. A., Connors, A., Drake, J. J., Izem, R., Meng, X. L., Min, S., Park, T., Ratzlaff, P., Siemiginowska, A., and Zezas, A. (2011). Accounting for calibration uncertainties in X-ray analysis: Effective areas in spectral fitting. *The Astrophysical Journal* **731**, 126–144.

Liang, F. (2010). A double metropolis–hastings sampler for spatial models with intractable normalizing constants. *Journal of Statistical Computation and Simulation* **80**, 9, 1007–1022.

Lira, I. (2002). *Evaluating the measurement uncertainty: fundamentals and practical guidance.* CRC Press.

Lpez-Garca, M. A., Lpez-Santiago, J., Albacete-Colombo, J. F., Prez-Gonzlez, P. G., and de Castro, E. (2013). A comprehensive study of ngc 2023 with xmm-newton and spitzer. *Monthly Notices of the Royal Astronomical Society* **429**, 775–791.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (2004). Equation of state calculations by fast computing machines. *The journal of chemical physics* **21**, 6, 1087–1092.

Moffat, R. J. (1988). Describing the uncertainties in experimental results. *Experimental thermal and fluid science* **1**, 1, 3–17.

Murray, I., Ghahramani, Z., and MacKay, D. (2012). Mcmc for doubly-intractable distributions. *arXiv preprint arXiv:1206.6848* .

Nevalainen, J., David, L., and Guainazzi, M. (2010). Cross-calibrating x-ray detectors with clusters of galaxies: an iachec study. *arXiv preprint arXiv:1008.2102* .

Park, T., van Dyk, D. A., and Siemiginowska, A. (2008). Searching for narrow emission lines in X-ray spectra: Computation and methods. *The Astrophysical Journal* **688**, 807–825.

Pollock, A. (2007). A new paradigm for the x-ray emission of o stars from xmm-newton observations of the o9.7 supergiant ζ orionis. *Astronomy & Astrophysics* **463**, 1111–1123.

Raassen, A. J. J., van der Hucht, K. A., Miller, N. A., and Cassinelli, J. P. (2008). Xmm-newton observations of ζ orionis (o9.7 ib): a collisional ionization equilibrium model. *Astronomy & Astrophysics* **478**, 513–520.

Ramsay, J. O. (2006). *Functional data analysis*. Wiley Online Library.

Rubin, D. B. (1987). The calculation of posterior distributions by data augmentation: Comment: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The sir algorithm. *Journal of the American Statistical Association* 543–546.

Siemiginowska, A., LaMassa, S., Aldcroft, T. L., Bechtold, J., and Elvis, M. (2008). X-ray properties of the gigahertz peaked and compact steep spectrum sources. *The Astrophysical Journal* **684**, 2, 811.

van Dyk, D. A., Connors, A., Kashyap, V., and Siemiginowska, A. (2001). Analysis of energy spectra with low photon counts via Bayesian posterior simulation. *The Astrophysical Journal* **548**, 224–243.

van Dyk, D. A. and Jiao, X. (2014). Metropolis-Hastings within partially collapsed Gibbs samplers. *Journal of Computational and Graphical Statistical (submitted)* .

van Dyk, D. A. and Park, T. (2008). Partially collapsed Gibbs samplers: Theory and methods. *Journal of the American Statistical Association* **103**, 790–796.

Waldron, W. L. and Cassinelli, J. P. (2001). Chandra discovers a very high density x-ray plasma on the o star ζ orionis. *The Astrophysical Journal* **548**, L45–L48.

Weise, K. and Woger, W. (1993). A bayesian theory of measurement uncertainty. *Measurement Science and Technology* **4**, 1, 1.

Yamauchi, S., Kamimura, R., and Koyama, K. (2000). Asca x-ray observations of the ngc 2023 and ngc 2024 regions. *Publications of the Astronomical Society of Japan* **52**, 1087–1096.

Appendix A

pyBLoCXS

Chandra Interactive Analysis of Observations (CIAO) is a data analysis system written for the needs of users of the Chandra X-ray Observatory. One of the central tools in CIAO system is the modeling and fitting package, Sherpa. It can be used for analysis of images, spectra and time series from many telescopes, including optical telescopes such as Hubble. It enables the user to construct complex models from simple definitions and fit those models to data, using a variety of statistics and optimization methods. Sherpa is an importable module for the Python scripting language, providing users with the ability to write their own Python scripts for use in Sherpa.

The `PyBLoCXS` is a sophisticated Markov chain Monte Carlo (MCMC) based algorithm designed to carry out Bayesian Low-Count X-ray Spectral (BLoCXS) analysis in the Sherpa environment. The code is a Python extension to Sherpa that explores parameter space at a suspected minimum using a predefined Sherpa model to high-energy X-ray spectral data. `PyBLoCXS` includes a flexible definition of priors and allows for variations in the calibration information. It can also be used to compute posterior predictive p-values for the likelihood ratio test.

MCMC is a complex computational technique that requires some sophistication on the part of its users to ensure that it both converges and explores the posterior distribution properly. The PyBLoCXS code has been tested with a number of simple single-component spectral models. It should be used with great care in more complex settings. Readers interested in Bayesian low-count spectral analysis should consult van Dyk *et al.* (2001). PyBLoCXS is based on the methods in van Dyk *et al.* (2001) but employs a different MCMC sampler than is described in that article. In particular, PyBLoCXS has two sampling modules. The first uses a Metropolis-Hastings jumping rule that is a multivariate t-distribution with user specified degrees of freedom centered on the best spectral fit and with multivariate scale determined by the Sherpa function, covar(), applied to the best fit. The second module mixes this Metropolis Hastings jumping rule with a Metropolis jumping rule centered at the current draw, also sampling according to a t-distribution with user specified degrees of freedom and multivariate scale determined by a user specified scalar multiple of covar() applied to the best fit.

For this thesis perspective, PyBLoCXS is used to sample θ from $p(\theta|A, R, Y)$. PyBLoCXS fixes the effective area curve and redistribution matrix, allowing no calibration uncertainty. Thus for simplicity, we use the notation $p(\theta|Y)$ as $p(\theta|A, R, Y)$.

Here we list the PyBLoCXS sampler briefly, discarding unrelated details to this thesis.

PyBLoCXS Sampler :

```
Step 0: conduct standard Sherpa fit(), get the best fit $\hat{\theta}$ as the starting value in Step 1

        use covar(), get the covariance matrix $\Sigma_0$ for the proposal setting

For $t = 0, 1, ..., T$

Step 1: Randomly sample $u_1$ from standard uniform distribution.
```

If $u_1 < p_m^{pyb}$, go to Step 1B, Metropolis Sampler

else, go to Step 1C, Metropolis-Hasting (Independence) Sampler

Step 2: Sample $\theta^{\text{prop}} \sim \mathcal{T}(df = df^{pyb}, \mu = \theta^{(t)}, \Sigma = scale^{pyb} * \Sigma_0)$

$$\alpha = \frac{p(\theta^{\text{prop}}|Y)}{p(\theta^{(t)}|Y)}$$

go to Step 1D

Step 3: Sample $\theta^{\text{prop}} \sim \mathcal{T}(df = df^{pyb}, \mu = \hat{\theta}, \Sigma = scale^{pyb} * \Sigma_0)$

$$\alpha = \frac{p(\theta^{\text{prop}}|Y)Q(\theta^{(t)})}{p(\theta^{(t)}|Y)Q(\theta^{\text{prop}})},$$

$Q(\theta^{\text{prop}})$ is the jumping rule, and here is t-distribution

go to Step 1D

Step 4: Randomly sample u_2 from standard uniform distribution.

$$\theta^{(t+1)}) = \begin{cases} \theta^{\text{prop}} & \text{if } u_2 < \alpha \\ \theta^{(t)} & \text{otherwise} \end{cases}.$$

where $\mathcal{T}(df, \mu, \Sigma)$ represents the t-distribution, df is the degree of freedom, μ is the center (mean), and Σ is the covariance matrix. Using the t-distributon as the proposal distribution instead of the normal distribution helps PyBLoCXS have broader choice of proposal distributions. When the target distribution $p(\theta|Y)$ has complicated structure (e.g. two-modal distribution), t-distribution can have better sampling performance since it has heavier tails than the normal distribution.

Therefore, the PyBLoCXS sampler has three important tuning parameters, p_m^{pyb}, df^{pyb} and $scale^{pyb}$. The default setting is $p_m^{pyb} = 0.5$, $df^{pyb} = 4$ and $scale^{pyb} = 1$. In complicated cases (e.g. very large data set), the tuning parameters setting can be crucial to the performance of the Monte Carlo chains. For example, if the target distribution has a very peaky mode, df^{pyb} should be set to be large, p_m^{pyb} to be close to one and $scale^{pyb}$ to be relatively small in order to obtain better convergence rate. The optimal tuning parameter setting heavily depends on the situation of the dataset.

In addition, `PyBLoCXS` also allows to build different types of prior distributions for θ, which may also influence the performance of the Monte Carlo chain. We skip the discussion about this here. More information about `PyBLoCXS` can be found at:

`http://hea-www.harvard.edu/astrostat/pyblocxs/`