2011 International Conference on Telecommunication Technology and Applications
Proc .of CSIT vol.5 (2011) © (2011) IACSIT Press, Singapore

Change-Point Modelling in Biological Sequences via the Bayesian Adaptive Independent Sampler

Georgy Sofronov [1] +

[1] Department of Statistics, Macquarie University, Sydney NSW 2109 Australia

Abstract. The change-point problem arises in wide variety of fields, including biomedical signal processing, speech and image processing, seismology, industry (e.g., fault detection) and financial mathematics. Multiple change-point models are also important in many biological applications and, particularly, in analysis of biomolecular sequences. We model genome sequences as a multiple change-point process, that is, a process in which the sequential data are separated into segments by an unknown number of change-points, with each segment supposed to have been generated by a different process. The parameters of the model are estimated by Adaptive Independent Samplers, which are adaptive Markov chain Monte Carlo methods based on the Independent Metropolis-Hastings algorithm. We discuss results of numerical experiments comparing different computing schemes.

Keywords: Markov chain Monte Carlo, adaptive methods, multiple change-point problem, comparative genomics

1. Introduction

The genomes of complex organisms, including the human genome, are known to vary in GC content along their length. That is, they vary in the local proportion of the nucleotides G and C, as opposed to the nucleotides A and T. Changes in GC content are often abrupt, producing well-defined regions.

We model DNA sequences as a multiple change-point process in which the sequence is separated into segments by an unknown number of change-points, with each segment supposed to have been generated by a different process. Multiple change-point problems are important in many biological applications, particularly in the analysis of DNA sequences. Multiple change-point problems also arise in segmentation of protein sequences according to hydrophobicity.

In this paper, we present an Adaptive Independent Sampler [6] to change-point modelling using Monte Carlo simulation to find estimates of change-points as well as parameters of the process on each segment. We include results of numerical experiments indicating the usefulness of this method. We obtain estimates for the locations of change-points in artificially generated sequences and compare the accuracy of these estimates to those obtained via Markov chain Monte Carlo [5] and the Cross-Entropy [1].

2. The Multiple Change-Point Problem

Let us formulate the multiple change-point problem in mathematical terms. A binary sequence $b = (b_1, ..., b_L)$ of length L is given. A segmentation of the sequence is specified by giving the number of change-points N and the positions of the change-points $c = (c_1, ..., c_N)$, where $0 = c_0 < c_1 < ... < c_N < c_{N+1} = L$. In this context, a change-point is a boundary between two adjacent segments, and the value c_n is the sequence position of the rightmost character of the segment to the left of the n-th change-point. A maximum number of change-points $N_{\max}$ is specified, where $0 \le N \le N_{\max} < L$. The model for the data assumes that within each segment characters are generated by independent Bernoulli trials with probability of success (that is obtaining a "1") θ

+ Corresponding author. Tel.: + 61 2 98508544; fax: +61 2 98507669.
E-mail address: georgy.sofronov@mq.edu.au.

that depends on the segment. Thus, the joint probability density of $b_1, ..., b_L$, conditional on N, $c=(c_1, ..., c_N)$, and $\theta = (\theta_0, ..., \theta_N)$, is given by

$$f(b_1, ..., b_L \mid N, c, \theta) = \prod_{n=0}^{N} \theta_n^{I(c_n, c_{n+1})}(1-\theta_n)^{O(c_n, c_{n+1})},$$

where

$$I(c_n, c_{n+1}) = \sum_{i=c_n+1}^{c_n+1} b_i, \quad O(c_n, c_{n+1}) = c_{n+1} - c_n - I(c_n, c_{n+1}).$$

In other words, $I(c_n, c_{n+1})$ is the number of ones in the segment bounded by sequence positions $c_n + 1$ and c_{n+1}, and $O(c_n, c_{n+1})$ is the number of zeros in that same segment.

To formulate the problem in terms of a Bayesian model, a prior distribution must be defined on the set of possible values of $\mathbf{z} = (N, c, \theta)$, denoted

$$Z = \bigcup_{N=0}^{N_{\max}} \{N\} \times C_N \times (0,1)^{N+1},$$

with

$$C_N = \{(c_1, ..., c_N) \in \{1, ..., L - 1\}^N : c_1 < ... < c_N\}.$$

We assume a uniform prior both on the number of change-points and on C_N, and uniform priors on (0,1) for each θ_n. Thus, the overall prior $f_0(N, c, \theta)$ is a constant. The posterior density at point $\mathbf{z} = (N, c, \theta)$, having observed $b_1, ..., b_L$, is thus given by

$$\pi(\mathbf{z}) \propto \prod_{n=0}^{N} \theta_n^{I(c_n, c_{n+1})}(1-\theta_n)^{O(c_n, c_{n+1})}.$$

3. The Bayesian Adaptive Independent Sampler

We consider the Independence Sampler (the simplest Metropolis-Hastings algorithm [3, 7]) to sample from a target density function π. When the proposal g is equal to π, the acceptance ratio is one and the correlation between adjacent elements of the chain is zero. Thus it is desirable to choose the proposal distribution g as closely as possible to the target π. In an adaptive framework, a natural strategy is to adjust a trial proposal g_0 to obtain a new proposal g that is 'closer' to π.

In this section, we describe the Bayesian Adaptive Independent Sampler [6], which involves running multiple parallel chains with a common proposal distribution. The proposal is updated by fitting a Bayesian model to the population of current elements across all chains. Updating the proposal is shown to have no effect on the limiting distribution of each chain, and thus can be continued indefinitely.

Let π be the target probability density function and let $g(x \mid \varphi)$ be a proposal distribution, defined up to a parameter φ, which is to be updated. Let $\varphi_0, \varphi_1, ...$ be the parameters for the sequence of proposals. Suppose we have K parallel chains $\{X_{1,j}, j = 1, 2, ...\}, ..., \{X_{K,j}, j = 1, 2, ...\}$, which we refer to as the sampling chains. Using the set of current elements $(X_{1,j}, ..., X_{K,j})$, we update φ_j at each step of the algorithm. That is, after updating each of the K chains, we update the proposal itself. Thus in effect we cycle through updates for K+1 chains, since $\varphi_0, \varphi_1, ...$ may also be regarded as values of an underlying Markov chain, which we refer to as the parameter chain.

In order to describe the algorithm more precisely and to show that the limiting distribution of each of the K sampling chains is the target distribution, we shall use a Metropolis-Within-Gibbs scheme, which can be considered within the framework of the Generalized Markov Sampler [4].

Let X denote the target space, that is, the space on which the target distribution π is defined. Let Φ denote the space of parameters for the proposal distribution. We may regard the K+1 parallel chains as a single chain defined on a space $\Phi \times X^K$.

Then the algorithm cycles through $(\varphi, x_1, ..., x_K)$ using a Metropolis-Within-Gibbs scheme. This scheme involves iterating two steps, known as the Gibbs step and the Metropolis-Hastings step. The Gibbs step is used to propose either a new element $y \in X$ or a new parameter $\varphi \in \Phi$. The Metropolis-Hastings step is used to accept or reject it in accordance with an acceptance probability. Both steps can be easily shown to satisfy detailed balance condition.

We consider the special case $X = R^d$ for some positive integer d. Then we may use a multivariate normal distribution as our proposal:

$$g(x \mid \varphi) = \mathbf{N}(x \mid \mu, \Sigma) \propto |\Sigma|^{-1/2} \exp\{-½\, (x - \mu)^{\mathrm{T}}\, \Sigma^{-1}\, (x - \mu)\}.$$

Using a non-informative prior, as in [2], we obtain the posterior distribution:

$$h(\varphi \mid x_1, \ldots, x_K) = h(\mu, \Sigma \mid x_1, \ldots, x_K) = \mathbf{N}(\mu \mid \bar{x}, \Sigma/K) \cdot \mathbf{Inv\text{-}W}_{K\text{-}1}(\Sigma \mid S),$$

where

$$S = \sum_{k=1}^{K} (x_k - \bar{x})(x_k - \bar{x})^T \ , \ \ \bar{x} = \sum_{k=1}^{K} x_k \,/\, K \, .$$

To obtain parameters μ and Σ, we first draw Σ from an Inverse-Wishart distribution $\mathbf{Inv\text{-}W}_{K\text{-}1}(\Sigma \mid S)$, then we draw μ from a Normal distribution $\mathbf{N}(\mu \mid \bar{x}, \Sigma/K)$. For further details, see [2].

Thus, the algorithm consists of the following steps performed iteratively:

Algorithm

Given $(x_1, \ldots, x_K)$, i, and $\varphi = (\mu, \Sigma)$:

1. Put

$$i' = \begin{cases} i+1, & \text{if } i \in \{0,1,\ldots,K-1\} \\ 0, & \text{if } i = K. \end{cases}$$

Generate $Y \sim \mathbf{N}(y \mid \mu, \Sigma)$ if $i' \in \{1,2,\ldots,K\}$. Generate $Y \sim$ \sim $\mathbf{N}(\mu \mid \bar{x}, \Sigma/K) \cdot \mathbf{Inv\text{-}W}_{K\text{-}1}(\Sigma \mid S)$ if $i' = 0$.

2. If $i' \in \{1,2,\ldots,K\}$, generate $U \sim \mathbf{U}(0,1)$ and deliver

$$x_{i'} = \begin{cases} Y, & \text{if } U \le \alpha_{i'}(x_{i'}, Y) \\ x_{i'}, & \text{otherwise,} \end{cases}$$

where

$$\alpha_{i'}(x_{i'}, y) = \min\{\rho_{i'}(x_{i'}, y), 1\} \, ,$$

with

$$\rho_{i'}(x_{i'}, y) = (\pi(y) \cdot \mathbf{N}(\mu \mid \bar{x}_y, \Sigma/K) \cdot \mathbf{Inv\text{-}W}_{K\text{-}1}(\Sigma \mid S_y) \cdot \mathbf{N}(x_{i'} \mid \mu, \Sigma))$$

$$\times\, (\pi(x_{i'}) \cdot \mathbf{N}(\mu \mid \bar{x}, \Sigma/K) \cdot \mathbf{Inv\text{-}W}_{K\text{-}1}(\Sigma \mid S) \cdot \mathbf{N}(y \mid \mu, \Sigma))^{-1},$$

$$S = \sum_{k=1}^{K} (x_k - \bar{x})(x_k - \bar{x})^T \ , \ \ \bar{x} = \sum_{k=1}^{K} x_k \,/\, K \, ,$$

$$S_Y = (x_1 - \bar{x}_Y)(x_1 - \bar{x}_Y)^T + \ldots + (x_{i'-1} - \bar{x}_Y)(x_{i'-1} - \bar{x}_Y)^T + (Y - \bar{x}_Y)(Y - \bar{x}_Y)^T$$

$$+ (x_{i'+1} - \bar{x}_Y)(x_{i'+1} - \bar{x}_Y)^T + \ldots + (x_K - \bar{x}_Y)(x_K - \bar{x}_Y)^T \, ,$$

$$\bar{x}_Y = (x_1 + \ldots + x_{i'\text{-}1} + Y + x_{i'+1} + \ldots + x_K)/K.$$

If $i' = 0$, put $(\mu, \Sigma) = Y$.

4. Results

In this section, we include results of numerical experiments that illustrate the performance of the BAIS. We consider an artificial sequence with a known distribution (see [1, 8]), which allows direct comparison with existing methods in terms of the Mean Squared Error (MSE).

Let $(b_1, b_2, \ldots, b_{22000})$ be a sequence of independent Bernoulli random variables generated with the parameters given in Table 1. The true profile of this sequence can be seen in Figure 1.

We generate 200 random sequences using these parameters and for each we run two variants of the BAIS:

- The BAIS that is used to determine the maximum of π, attained at the "best model". We denote this method BAIS1.
- The BAIS that takes the average of the sample produced by K parallel chains. This method is denoted by BAIS2.

We consider a 21-dimensional normal proposal distribution $\mathbf{N}(\mu, \Sigma)$. We choose initial values μ_i for the mean vector μ such that each μ_i is equally spaced over the set $\{0, \ldots, L\}$ and $\Sigma = (500, \ldots, 500, 0.25, \ldots, 0.25)$

$\cdot\ I_{21}$, where I_{21} is the 21×21identity matrix. In fact, the limiting distribution is independent of the initial values of μ and Σ, and they should not greatly affect the sampler efficiency. We run the algorithm for 3000 iterations, using $K = 50$ parallel chains.

Recall that $N_{\max}$ is the maximum number of change-points we wish to find. We can represent the positions of the change-points as a non-decreasing $N_{\max}$-dimensional vector. When the number of change-points is less than $N_{\max}$, the value of some components in the vector will be repeated indicating the "same" change-point.

The algorithm produces a single vector of estimates of the positions of change-points and the Bernoulli parameters. A GC profile is a vector of length L where for each $i \in \{1, \ldots, L\}$, GC(i) is the average GC content in the segment containing the ith character in the sequence. A GC profile is produced from the change-point vector as follows:

$$\mathrm{GC}(i) = \theta_j \text{ where } \mathrm{c}_{j-1} < i \leq c_j \text{ and } j = 1, \ldots, N_{\max}.$$

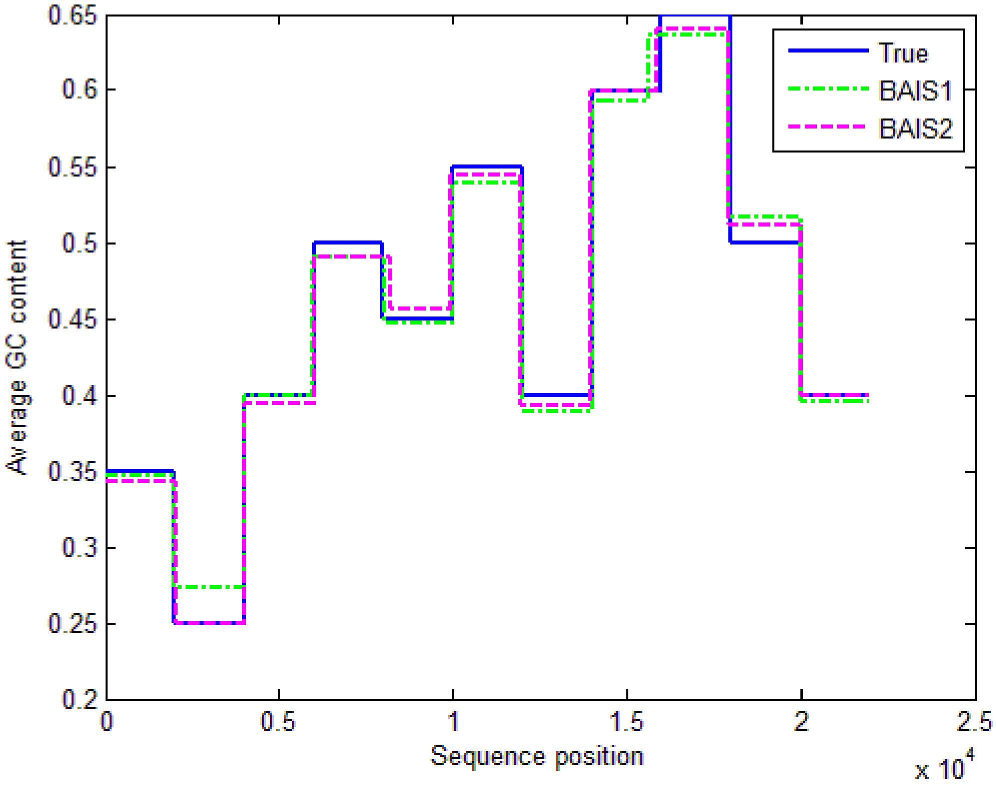

Fig. 1: A profile plot comparing the average GC content along the full length of the artificial sequence as determined by the BIAS1 and the BIAS2 to the true profile.

Table 1: Bernoulli parameters for artificial sequence.

Positions	*Bernoulli parameter*
1 - 2,000	$\theta_0 = 0.35$
2,001 – 4,000	$\theta_1 = 0.25$
4,001 – 6,000	$\theta_2 = 0.4$
6,001 – 8,000	$\theta_3 = 0.5$
8,001 – 10,000	$\theta_4 = 0.45$
10,001 – 12,000	$\theta_5 = 0.55$
12,001 – 14,000	$\theta_6 = 0.4$
14,001 – 16,000	$\theta_7 = 0.6$
16,001 – 18,000	$\theta_8 = 0.65$
18,001 – 20,000	$\theta_9 = 0.5$
20,001 – 22,000	$\theta_{10} = 0.4$

The profiles for the BIAS1 and the BIAS2 from a single random sequence can be seen in Figure 1. These two plots are in excellent agreement with each other, supporting the fact that both produced only very small difference between their estimates and the true distribution. It is interesting to note that both algorithms almost always under-estimated or over-estimated the GC content in the same direction. This could be attributed to the fact that although the artificial sequence was drawn from Bernoulli random variables with parameters given in Table 1, it is still a random process and it is possible that the true GC proportions are not exactly the same as the Bernoulli parameters of Table 1.

To determine the quality of the profiles, the MSE is calculated as

$$\text{MSE} = \sqrt{\sum_{i=1}^{22000} (t(i) - e(i))^2} \ ,$$

where $t(i)$ is the true GC proportion and $e(i)$ is the estimated GC proportion at position i. In order to compare the BAIS to other methods, we consider the following algorithms:

- The MCMC approach [5] with 100 samples with a step size of 3000.
- The Cross-Entropy (CE) method [1] with a sample size of 1000, smoothing parameters of 0.7 and 0.3 and an elite proportion value ρ of 0.01.
- The Sequential Importance Sampling (SIS) algorithm [8] with $d = 10$, $N_1 = 500$.

The results are displayed in Table 2.

Table 2: The average Mean Squared Error.

Algorithm	*MSE*
BIAS1	3.6
BIAS2	3.3
MCMC	3.0
CE	3.4
SIS	4.4

It illustrates that the BAIS performs well in comparison to the other methods. The MSE for the BIAS is somewhat larger than, but still comparable to, the MCMC approach [5], which uses a much more difficult and computationally intensive segmentation method.

5. References

[1] G. E. Evans, G. Yu. Sofronov, J. M. Keith, D. P. Kroese. Estimating Change-Points in Biological Sequences via the Cross-Entropy Method. *Annals of Operations Research*. DOI 10.1007/s10479-010-0687-0.

[2] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Rubin. *Bayesian Data Analysis*. Chapman and Hall, London, second edition, 2003.

[3] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*. 1970, **57**: 97-109.

[4] J. Keith, D. P. Kroese, D. Bryant. A generalized Markov sampler. *Methodology and Computing in Applied Probability*. 2004, **6** (1): 29-53.

[5] J .M. Keith, P. Adams, S. Stephen, J. S. Mattick. Delineating slowly and rapidly evolving fractions of the Drosophila genome. *Journal of Computational Biology.* 2008, **15** (4): 407-430.

[6] J. M. Keith, D. P. Kroese, G. Yu. Sofronov. Adaptive Independence Samplers. *Statistics and Computing*. 2008, **18** (4): 409-420.

[7] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller. Equations of state calculations by fast computing mashines. *Journal of Chemical Physics*. 1953, **21**: 1087-1092.

[8] G. Yu. Sofronov, G. E. Evans, J. M. Keith, D. P. Kroese. Identifying Change-points in Biological Sequences via Sequential Importance Sampling. *Environmental Modeling and Assessment.* 2009, **14** (5): 577-584.