Mechanical Systems and Signal Processing 93 (2017) 379–396

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

Estimating the parameters of dynamical systems from Big Data using Sequential Monte Carlo samplers

P.L. Green [a,c,*], S. Maskell [b,c]

[a] School of Engineering, University of Liverpool, Liverpool L69 7ZF, UK
[b] Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 7ZF, UK
[c] Institute for Risk and Uncertainty, University of Liverpool, Liverpool L69 7ZF, UK

ARTICLE INFO

Article history:
Received 29 April 2016
Received in revised form 16 December 2016
Accepted 19 December 2016

Keywords:
Big Data
Parameter estimation
Model updating
System identification
Sequential Monte Carlo sampler

ABSTRACT

In this paper the authors present a method which facilitates computationally efficient parameter estimation of dynamical systems from a continuously growing set of measurement data. It is shown that the proposed method, which utilises Sequential Monte Carlo samplers, is guaranteed to be fully parallelisable (in contrast to Markov chain Monte Carlo methods) and can be applied to a wide variety of scenarios within structural dynamics. Its ability to allow *convergence* of one's parameter estimates, as more data is analysed, sets it apart from other sequential methods (such as the particle filter).

1. Introduction

This paper addresses the situation where one is attempting to infer the parameters of a dynamical model from a large set of data which, because of its size, cannot be processed using current methods. Here, $\boldsymbol{z}_t$ denotes a vector of measurements, obtained at time t, and $\boldsymbol{\theta}$ is a vector of the model's parameters. The aim is to realise probabilistic estimates of $\boldsymbol{\theta}$, given the available data, via Bayes' theorem:

$$p(\boldsymbol{\theta}|\boldsymbol{z}_{1:n}) \propto p(\boldsymbol{z}_{1:n}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{1}$$

where $\boldsymbol{z}_{1:n} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ represents the set of all measurements up to time $t = n$. In [1,2] it was suggested that, using Markov chain Monte Carlo (MCMC) methods, one could generate samples from $p(\boldsymbol{\theta}|\boldsymbol{z}_{1:t})$ while t is gradually increased. Such an approach facilitates a gradual transition from prior to posterior, which aids MCMC convergence (in a similar manner to simulated annealing). It also allows one to analyse how one's parameter estimates converge as more data is analysed, thus helping to establish when a sufficient amount of data has been utilised. The computational cost of such an approach, however, increases dramatically as more data is analysed. This makes it poorly suited to the situation where large sets of new (potentially important) measurements are expected to arrive in the future. Other approaches such as [3] involve the selection of small subsets of 'highly informative' training data from large data sets. While this reduces computational cost, it involves the deliberate omission of measurement data which, in hindsight, may contain important information (and reduce uncertainty in the posterior as a result).

* Corresponding author at: School of Engineering, University of Liverpool, Liverpool L69 7ZF, UK.

http://dx.doi.org/10.1016/j.ymssp.2016.12.023
0888-3270/

In this paper, an algorithm based on Sequential Monte Carlo (SMC) methods is proposed, which is able to address the aforementioned issues. Fundamentally, the efficiency of the method proposed here lies in its ability to exploit the inevitable redundancies that arise in large sets of measurements, as well as its suitability for modern computing architectures. It is important to note that the method proposed in this paper is different from other, recently proposed sampling methods ([4,5] for example), as it is specifically aimed at the situation where a prohibitively large data set is available. The proposed method also allows one to track how the uncertainties in one's parameter estimates reduce as more data is analysed - thus establishing when a sufficient amount of data has been processed.

In the interest of completeness, a brief introduction to SMC methods, as well as a description of previous work relevant to the problem of interest, is given in the following section.

2. Sequential Monte Carlo methods

2.1. Importance sampling

This section begins with a brief description of importance sampling. Here $\pi(\boldsymbol{\theta})$ is defined as a target probability distribution, from which one wishes to estimate the expected value of a function, $f(\boldsymbol{\theta})$. $\pi^*(\boldsymbol{\theta})$ is used to represent an unnormalised target, such that:

$$\pi(\boldsymbol{\theta}) = \frac{\pi^*(\boldsymbol{\theta})}{Z}, \qquad Z = \int \pi^*(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{2}$$

(for generality it is assumed that Z is difficult to estimate here - a situation which often arises in Bayesian inference problems). The expected value of $f(\boldsymbol{\theta})$ can be written as

$$\mathrm{E}[f(\boldsymbol{\theta})] = \frac{\int f(\boldsymbol{\theta})\pi^*(\boldsymbol{\theta}) d\boldsymbol{\theta}}{\int \pi^*(\boldsymbol{\theta}) d\boldsymbol{\theta}} = \frac{\int f(\boldsymbol{\theta}) q(\boldsymbol{\theta}) w(\boldsymbol{\theta}) d\boldsymbol{\theta}}{\int q(\boldsymbol{\theta}) w(\boldsymbol{\theta}) d\boldsymbol{\theta}} \tag{3}$$

where $w(\boldsymbol{\theta}) = \frac{\pi^*(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}$ are 'importance weights' and $q(\boldsymbol{\theta})$ is a user-defined 'proposal distribution' - a probability distribution from which it is relatively easy to generate samples. Eq. (3) implies that

$$\mathrm{E}[f(\boldsymbol{\theta})] \approx \sum_{i=1}^{N} f(\boldsymbol{\theta}^i) \tilde{w}^i \tag{4}$$

where $\{\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^N\}$ have been generated from $q(\boldsymbol{\theta})$ and, adopting the notation $w^i \equiv w(\boldsymbol{\theta}^i)$,

$$\tilde{w}^i = \frac{w^i}{\sum_j w^j}, \qquad i = 1, \ldots, N \tag{5}$$

are defined as 'normalised importance weights'. This reweighting procedure allows estimates of $\mathrm{E}[f(\boldsymbol{\theta})]$ to be realised using samples from $q(\boldsymbol{\theta})$, which is useful when it is difficult to generate samples from the target distribution, $\pi(\boldsymbol{\theta})$, directly.

2.2. Resampling

By defining $f(\boldsymbol{\theta}_j) = \delta(\boldsymbol{\theta}_j - \boldsymbol{\theta})$ where δ is the Dirac delta function, it follows that

$$\mathrm{E}[f(\boldsymbol{\theta}_j)] = \int \delta(\boldsymbol{\theta}_j - \boldsymbol{\theta}) \pi(\boldsymbol{\theta}) d\boldsymbol{\theta} = \pi(\boldsymbol{\theta}_j). \tag{6}$$

This implies that, if one has a set of samples (and accompanying normalised weights) $\{\boldsymbol{\theta}^1, \tilde{w}^1\}, \ldots, \{\boldsymbol{\theta}^N, \tilde{w}^N\}$ while a new set of samples, $\{\bar{\boldsymbol{\theta}}^1, \ldots, \bar{\boldsymbol{\theta}}^N\}$, is chosen such that

$$\Pr(\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}^i) = \tilde{w}^i \tag{7}$$

then $\{\bar{\boldsymbol{\theta}}^1, \ldots, \bar{\boldsymbol{\theta}}^N\}$ will be approximate samples from the target. The weights of these new samples will therefore be equal (for more information the tutorial [6] is recommended). Resampling is often used when it is found that relatively few of the current samples have significant weight as it helps to remove those samples which are of little importance. It is often used to tackle the 'degeneracy' problem that can be encountered in the application of particle filters. To indicate when resampling is required, the concept of 'effective sample size' was introduced in [7,8]. This involves defining

$$N_{eff} = \frac{1}{\sum_i (\tilde{w}^i)^2} \tag{8}$$

and choosing to conduct resampling when N_{eff} falls below some kind of threshold ($N/2$, for example, is used throughout the current paper). It should be noted that, while resampling helps to remove 'unimportant' samples, it doesn't aid exploration of the parameter space - it can only produce replicas of the existing samples.