Abstract

.....

Introduction

Time series are widely used in applied science and technology. Time series database management systems are the fastest growing segment in the database industry, which may indicate a growing need for time series forecasting.

Often, time series data is heterogeneous and experiences multiple abrupt changes in structure. These changes are known as changepoints cause the data to be split into segments which can then be modelled separately. Changepoint detection is required in B a number of applications including financial data, bioinformatics, climate data and analysis of speech signals.

As increasingly large data-sets are obtained there is a need for statistical methods for detecting changepoints that are not only accurate but also are computationally efficient.

Many approaches to estimating the number and position of changepoints can be formulated in terms of defining a cost function for a segmentation. They either minimise a penalised version of this cost (the penalised minimisation problem) or minimise the cost under a constraint on the number of changepoints (the constrained minimisation problem).B

In the case where the cost function depends on the data through a sum of segment-specific costs, the minimisation can be done exactly B using dynamic programming. However the exact dynamic programming methods have a cost that increases at least quadratically with the amount of data.

We can speed up the dynamic programming algorithms by the pruning of the solution space. These algorithms are widely used the techniques "inequality based pruning"and "functional pruning".

The focus of this report is on using the Gaussian cost function andB the technique "functional pruning"to solve the penalised minimisation problem B for multivariate changepoint detection. We consider a time series of dimention $p \geq 2$. The case $p = 1$ corresponds to the classical univariate framework, left aside in this study.

We present a new approach to search for changepoints ...

We show that ...

The raport is organised as fallows. We begin in Section 1 by description of the penalised optimisation problems for data segmentation. We then rewiew the existing dynamic programming algorithms and pruning approaches for solving the penalised optimisation problem in Section 2. We introduce the new heuristic algorithms FPOP1, FPOP2, FPOP3 in Section 3 and compare these methods in Section 4.....

1 Multiple changepoint detection

1.1 Model definition

We consider a multivariate time series x is a vector of ordered data of dimension p and size n, that is $x = x_{1:n} = (x^1, .., x^p)_{1:n} \in (\mathbb{R}^p)^n$ with x_i^j the j-th component of the p-dimensional point $x_i \in \mathbb{R}^p$ in position i in vector x. The parameters p and n are satisfied the constraints $n \gg p$ and $p \geq 2$. Within this raport, we use the notation $x_{s:t} = (x_s, .., x_t)$.

Each data point is generated by p-parametric laws from the natural exponential family. Parameters of the laws are constant on each segment (common to all the p joint series). We assume an overall independance: components of each data point are independent as well as the successive data points.

The problem of "Multiple changepoint detection"is consisting in finding both the location of changepoints and the parameters associated to each segment and is related to the minimization of cost function linked to the data model in this parametric configuration. We suppose that the cost function $C()$ satisfies the condition (1) for all $s < t$ and some single observation cost function $\theta \to \omega(x_i, \theta)$ with observation $x_i \in \mathbb{R}^p$ and parameter $\theta \in \mathbb{R}^p$.

$$C(x_{s:t}) = \min_{\theta \in \mathbb{R}^p} \sum_{i=s}^{t} \omega(x_i, \theta). \tag{1}$$

This cost function is defined as the opposite of log-likelihood of the probability density function. In the case of a natural exponential family (Gaussian, Poisson, Binomial distributions) the cost is piecewise quadratic(Gaussian) or the sum of a linear term with a non-linear convex function (other cases).

The segmentation with k changepoints is defined by the vector $\tau = (1 = \tau_0, .., \tau_{k+1} = n+1)$. The segments are then given by the sets of indices $(\tau_i, \tau_{i+1} - 1), i = 0, .., k$. In order to find the optimal number of segments, we use a penalised version of the cost by $\beta > 0$. We define the set $S_n = \{\tau = (\tau_0, .., \tau_{k+1}) \in \mathbb{N}^{k+2} | 1 = \tau_0 < \tau_1 < .. < \tau_k < \tau_{k+1} = n + 1\}$, then the optimal cost associated to our segmentation problem is:

$$Q_n = \min_{\tau \in S_n} \sum_{i=0}^{k} \{C(x_{\tau_i:\tau_{i+1}-1}) + \beta\}. \tag{2}$$

Finally, the desired quantity is the argument of the previous minimisation. The optimal segmentation τ^* is then defined as:

$$\tau^* = \operatorname*{Arg\,min}_{\tau \in S_n} \sum_{i=0}^{k} \{C(x_{\tau_i:\tau_{i+1}-1}) + \beta\}. \tag{3}$$

1.2 Conditions for pruning

The pruning methods are used for speeding up the dynamic programming algorithms. The pruning methods can be applied under one of two conditions on the segment costs:

- **Condition1**: The functional pruning

  The cost function satisfies

$$C(x_{t+1:s}) = \min_{\theta \in \mathbb{R}^p} \sum_{i=t+1}^{s} \omega(x_i, \theta), \tag{4}$$

  for some function $\omega()$, with parameter θ.

- **Condition2**: The inequality based pruning

  There exists a constant k such that for $t < s < T$,

$$C(x_{t+1:s}) + C(x_{s+1:T}) + k \leq C(x_{t+1:T}). \tag{5}$$

Note: Condition1 is a stronger than Condition2. If Condition1 holds then Condition2 also holds with $k = 0$.

2 The exact dinamic programming algorithms for solving the penalised minimisation problem

At the beginning of this section we will discuss the Optimal Partitioning (OP) algorithm (Jackson et al., 2005). Then we will consider how pruning can be used to increase the computational efficiency of the OP-method whilst still ensuring that the method finds a global minimum of the cost function. The essence of pruning in this context is to remove those values of τ which can never be minima from the minimisation performed at each iteration of Optimal Partitioning method.

2.1 Optimal Partitioning

The idea of the OP-method is to split the minimisation over segmentations into the minimisation over the position of the last changepoint, and then the minimisation over the earlier changepoints. We introduce the parameter θ is the vector of last segment parameters within the data. We define

$$Q_{t+1}(\theta) = \min_{\tau \in S_{t+1}} \left[\sum_{i=0}^{k-1} \{C(x_{\tau_i:\tau_{i+1}-1}) + \beta\} + \{\sum_{j=\tau_k}^{t+1} \omega(x_j, \theta) + \beta\} \right]. \tag{6}$$

and obtain the recursion (7) for all $\theta \in \mathbb{R}^p$, with $m_t = \min_\theta Q_t(\theta)$ and the initialisation $Q_0(\theta) = 0$ (so that $Q_1(\theta) = \omega(x_1, \theta)$).

$$Q_{t+1}(\theta) = \min\{Q_t(\theta), m_t + \beta\} + \omega(x_{t+1}, \theta), \tag{7}$$

This recursion can be solved in turn for $t = 1 : n$. The cost of solving the recursion for time t is linear in t, so the overall computational cost of finding Q_n is quadratic in n. The last changepoint at the step t, $cp(t) = \tau_t$ in series $x_{1:t}$ is given by

$$\tau_t = \underset{1 \le i < t}{\text{Arg min}}\{Q_i(\theta) + \omega(x_{i+1:t}, \theta) + \beta\}. \tag{8}$$

The optimal changepoints up to time t can be calculated recursively $cp(t) = (cp(\tau_t), \tau_t)$.

2.2 PELT

One way to increase the efficiency of OP-method is the Pruned Exact Linear Time (PELT) algorithm (Killick et al.(2012)). PELT works by limiting the set of potential previous changepoints. If Condition2 (5) holds for some k, and if

$$Q_t(\theta) + C(x_{t+1:s}) + k > Q_s(\theta), \tag{9}$$

then at any future time $T > s$, t can never be the optimal location of the most recent changepoint prior to T. This means that at every time step s the left hand side of equation (9) can be calculated for all potential values of the last changepoint. If the inequality holds for any individual t then that t can be discounted as a potential last changepoints, τ to consider. This set , which we shall denote as

$$R_{t+1} = \{\tau \in \{R_t \cup \{t\}\} : Q_\tau(\theta) + C(x_{\tau+1:t}) + k \le Q_t(\theta)\}. \tag{10}$$

This pruning technique, which we shall refer to as "inequality based pruning forms the basis of the PELT-method.As at each time step in the PELT algorithm the minimisation is being run over fewer values it would be expectedthat this method would be more efficient than the basis OP algorithm. In Killick et al.(2012) it shown to be at least as efficient as OP-method, with PELT's computational cost being bounded above by $O(n^2)$. Under certain conditions the expected computational cost can be shown to be bounded by Ln for some constant $L < \infty$. These conditions are given fully in Killick et all.(2012), the most important of which is that the expected number of changepoints in the data increases linearly with the length of the data,n.

2.3 FPOP

3 Heuristic algorithms FPOP1, FPOP2, FPOP3

3.1 FPOP1

3.2 FPOP2

3.3 FPOP3

4 Simulation

5 Dimension 2. FPOP1, FPOP2, FPOP3

.....

5.1 A multi-dimentional Gaussian cost

In our study we use the criterio SIC for choice of the penalty β.

we will restrict our study to the natural exponential familly with a normal distribution.

Each data point is generated by p-parametric normal distribution law. Parameter μ is conctant on each segment and parameter σ is conctant for all series (common to all the p joint series).

Parameter μ is conctant on each segment and parameter σ is conctant for all series (common to all the p joint series).

The problem consisting in finding both the location of changepoints and the parameter μ associate to each segment is called multiple changepoint detection

The focus of this report is on the solving the penalised minimisation problem using a dinamic programming approach with the functional pruning.

The discrete minimisation of overall cost in (2) involves the scanning all the 2^{n+1} diffrent segmentations.

As equation (7) for time steps $t = 1 : n$ and each time step involves a minimisation over $\tau = 1 : t$ the computation actually takes $O(n^2)$ time.

Each function $Q_t()$ is a piecewise continuous function made of, at most, t different functions.

The segmentations themselves can recovered by first taking the arguments which minimise (7)