Bachelor's project

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

Amir Ahmed

Breakpoint detection on latent autoregressive time series of counts using integrated nested Laplace approximation.

Bachelor's project in mathematics (BMAT)

Supervisor: Jo Eidsvik

June 2020

NTNU
Norwegian University of Science and Technology

Abstract

Abrupt changes in a data source can weaken models that fail at addressing these. Structural change detection has traditionally been done with a frequentist approach, but recently approaches based on Bayesian models and Markov Chain Monte Carlo (MCMC) schemes have seen more use. The Integrated Nested Laplace Approximation (INLA) method was developed as a computationally efficient alternative to MCMC sampling. This text experiments with how the INLA approach can be applied in detecting breaks in time series of counts.

It is investigated how different metrics such as, marginal likelihood, comparison of posterior marginals with the L2 norm, and the Deviance Information Criterion (DIC) perform in detecting two types of breaks. The first break type is in correlation structure and the second break type is in the variance structure. The results show that marginal posterior likelihood and DIC perform best when correlation breaks, and that the L2 norm is the best metric of the three with variance structure change.

Lastly the methods presented in this text are used to detect break points in the correlation structure of trading volume data on the TSLA-stock. Two breakpoints were found.

Contents

1 Introduction

Parameters are often unstable in models with large predictor space. The data generating process might change with the predictors causing instability in the parameters. A goal of statistical modeling is to be able to predict accurately, and failing to address structural process changes would leave a model weaker. Changing parameters with partitions of the predictor space is a simple method for dealing with this issue. Instead of assuming stability of a parameter over the whole predictor space one rather assumes stability in each partition. Usually one

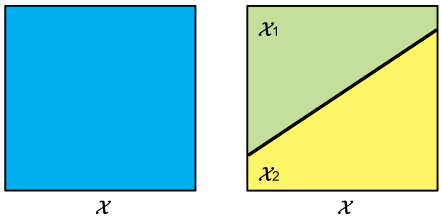


Figure 1: Display of how one can partition the predictor space for an arbitrary model $\mathcal{M}(\boldsymbol{\theta}, \mathcal{X})$ over the predictor space $\mathcal{X}$ opening for the use of different values for $\boldsymbol{\theta}$ in the two new partitions.

has predictor space $\mathcal{X}$ and finds optimal parameters $\boldsymbol{\theta}^*$ for the model $\mathcal{M}(\boldsymbol{\theta}^*, \mathcal{X})$ see Figure 1. However if there is some structural break dividing the predictor space into disjoint spaces $\mathcal{X}_1$ and $\mathcal{X}_2$ the true parameters might be $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ and not $\boldsymbol{\theta}^*$. Consequently basing inference on $\boldsymbol{\theta}^*$ might introduce unnecessary errors, and in cases with data prone to parameter instability being able to reliably detect such changes and find such partitions becomes important. A simple example could be modeling a trait of a specific species of a plant that changes behavior in different biomes. A natural solution would be to partition after biome.

This text will focus on investigating the following hypothesis, $\mathcal{H}_0$ that there is a break in the parameter structure versus $\mathcal{H}_1$ stating that there is no break. More specifically the focus will be on breakpoint detection in time series of counts. Partitioning the predictor space then means to find segments of time where the count process behaves similarly. A lot of work has already been done in structural break tests on time series, and breakpoint detection in time series has been widely studied in fields such as econometrics and signal analysis. Most research on time series of counts has been done in the recent years as evaluation of these types of models are more computationally intensive.

We take a Bayesian approach to model fitting, and will assume that the parameters of our models follow some distribution and adopt the use of the integrated nested Laplacian approximation (INLA) scheme for model fitting. It is assumed that all breaks happen at a distinct point in time. An alternative, however out of the scope of this text, would be to let breakpoints follow some probability distribution over the predictor space.

1.1 Current approaches to breakpoint detection

Methods in breakpoint detection often take a frequentist approach. For breakpoint detection in time series models, the cumulative sum (CUSUM) is one of the most well known approaches. It was presented in Page (1954) and the method is relatively flexible. Andrews (1993) and Tsay (1988) both develop statistics that can be used to test for breaks using CUSUM. Brown et al. (1975) develop the CUSUM and the CUSUM of squares test to find parameter instability in linear models, basing the tests on the residuals of the model fit. For more complex models tests based on ML-statistics are often used Hjort and Koning (2002) and Zeileis and Hornik (2007) both develop techniques using ML-estimates of parameters. Zeileis for instance applies the functional central limit theorem in order to show convergence of some score function to a Brownian bridge. An alternative is to take a Bayesian approach as Chen and Lee (2016) who uses MCMC and bases model choice and splitting criterion on the DIC statistic.

Time series of counts are widely dealt with in econometrics such as in Winkelmann (2008). There are several different techniques and model frameworks used in breakpoints detection for count processes. Lee et al. (2016) develop a CUSUM like residual test. Abujiya (2017) presents ways of transforming the intensity approximation in a count processes to something that is approximately normal. Doukhan and Kengne (2013) use a frequentist approach to find breakpoints in the INGARCH model, they use the likelihood connected to the maximum likelihood estimates of the parameters to split points and transform it to something asymptotically equivalent to a Brownian bridge. Chen and Lee (2016) take a different approach and applies MCMC with Metropolis Hastings to the ZIGP INGARCH model to detect breakpoints, with zero-inflation to handle over-dispersion.

The text has the following structure. Section 2.1 and Section 2.2 explain the model more in depth. Section 2.3 explains the INLA scheme. Section 3 and Section 4 explain and test a framework based on INLA to detect change points. To illustrate a possible application of the work in this text, real data is analyzed in Section 5, and the text tests for breakpoints in the correlation structure of the daily trading volume of the TSLA-stock[1] . The data is displayed in Figure 2.

2 Latent Gaussian models and INLA

2.1 Model specification

Among the simplest time series models is the autoregressive model of order 1, denoted AR(1). This text will use the following model for the AR(1),

$$X_1 = \epsilon_1 \sim N(\mu, \sigma^2(1-\rho^2)^{-1}), \quad X_t = \rho X_{t-1} + \epsilon_t \tag{1}$$

$$\epsilon_t \sim N(0, \sigma^2), \quad 2 \leq t \leq n.$$

Let $\mathbf{X} = (X_1, \ldots, X_n)^T$. To ensure stationarity we restrict $|\rho| < 1$. The variance of X_1 is set to give constant variance at all times. The notation $N(\cdot, \cdot)$ means

[1] Trading data based on Tesla, Inc. (TSLA) from Yahoo! Finance `https://finance.yahoo.com/quote/TSLA` retrieved 27/04/2020 from `https://www.kaggle.com/timoboz/tesla-stock-data-from-2010-to-2020`

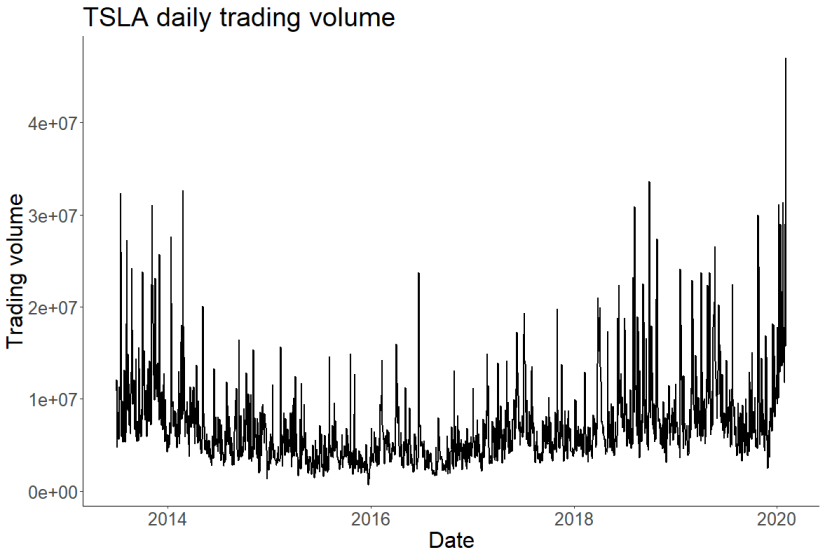


Figure 2: Display of the daily trading volume of the TSLA-stock. Trade volume is a count value.

a Gaussian variable with specified mean and variance. Noise terms, $\epsilon_1, \ldots, \epsilon_n$ are assumed to be zero-centered Gaussian and independently distributed. The time series X_t could for instance represent the change of the price of a stock at time $t > 0$ after some initial $t = 0$. The model would assume correlation to the day before, with some added noise.

The goal of this text is to be able to detect breakpoints, for the aforementioned model there could for instance exist some $1 < T < n$ that alters the model into,

$$X_{t+1} = \begin{cases} \rho_1 X_t + \epsilon_{1t}, & 1 < t \leq T \\ \rho_2 X_t + \epsilon_{2t}, & T < t \\ \epsilon_1, & t = 1, \end{cases} \tag{2}$$

where $\epsilon_{1t} \sim N(0, \sigma_1^2)$, $\epsilon_{2t} \sim N(0, \sigma_2^2)$ and $\epsilon_1 \sim N(\mu, \sigma_1^2(1-\rho_1^2)^{-1})$, again assuming $|\rho_1| < 1$ and $|\rho_2| < 1$. In the case of the stock-market, some event might have happened at T that changed the underlying price determining process. More breaks can be introduced in a similar fashion. Note that varying number of breakpoints would give different dimensional parameter space to ease notation a models parameters will be denoted with the following $\boldsymbol{\theta} = (\rho_1, \rho_2, \ldots, \sigma_1, \sigma_2, \ldots)$, the elements of $\boldsymbol{\theta}$ will be assumed to follow some independent distributions.

Figure 3a displays a time series without any breakpoints. Figure 3b illustrates a simulated case of (2) with a change in variance. Increasing the variance gives larger difference between consecutive observations. In this case the breakpoints segments the predictor space into two partitions. If one knows the value of T it is easy to build that into the model, but if its position is unknown picking a value for T might prove more difficult. Another possibility is that correlation of the time series varies, i.e. $\rho_{11} \neq \rho_{21}$. A simulated case of this is displayed in

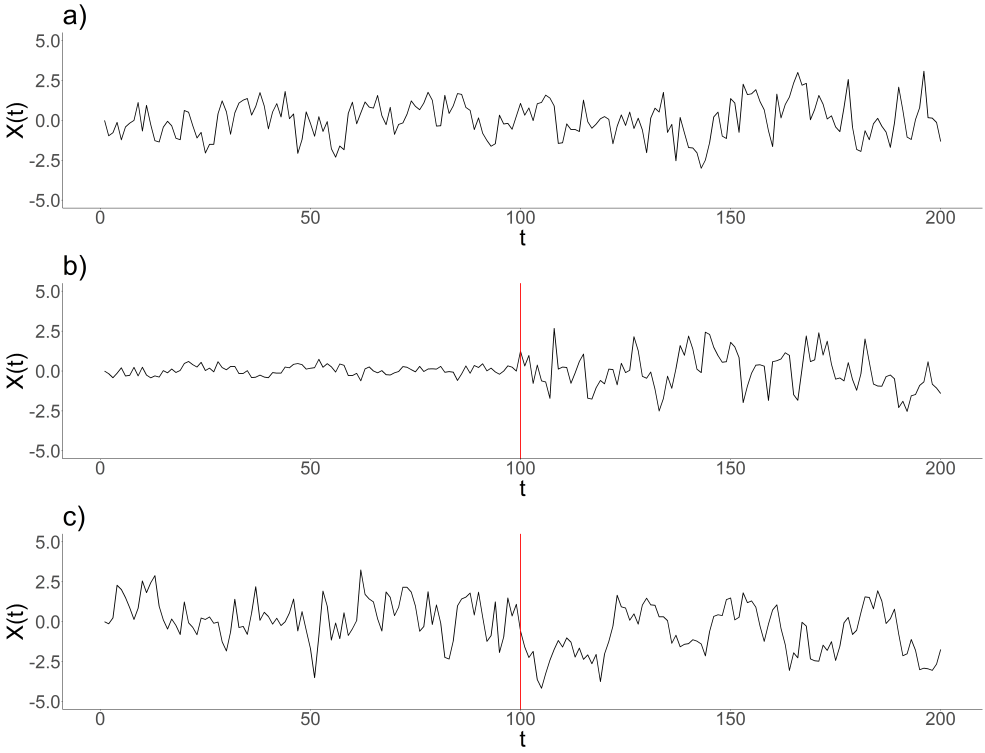


Figure 3: Different simulated AR(1) time series. a) No breaks $\rho = 0.5, \sigma = 1$. b) change at $T = 100$ in variance $\sigma_1 = 0.25$, $\sigma_2 = 1$, $\rho = 0.5$. c) change at $T = 100$ in correlation $\rho_1 = 0.5$, $\rho_2 = 0.8$, $\sigma = 1$.

Figure 3c. From Figure 3c we see that the increased correlation creates chunks with high and low centered mostly around 0.

In many cases, however, it is favorable to assume that such a temporal model is latent in some other process. This can be done when dealing with time series of counts. A model could for instance be,

$$Y_t|\lambda_t \sim Poisson(\lambda_t), \tag{3}$$

$$\lambda_t = \exp(X_t),$$

where X_t is modeled as in (1) and where $Y_1|\lambda_1, \ldots, Y_n|\lambda_n$ are independent and Poisson distributed. $Poisson(\cdot)$ denotes a Poisson distributed variable with a given intensity parameter. One says that the observations $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ are conditionally independent given the intensity in the Poisson process and the intensity is a latent AR(1) process. Here, the term, latent simply means that the AR(1) time series is not directly observed. The text focuses on methods to detect structural breaks in cases where one only observe $\mathbf{Y}$.

Figure 4 displays possible observations based on simulations of (3). Even though the time series is now wrapped in a Poisson distribution much of the same as observed in Figure 3. Increased correlation gives chunks of highs and lows, while increased variance gives larger differences and jumps between consecutive

days. Note that the display in Figure 3 is the latent time series of the respective realizations displayed in Figure 4. The model described in (3) has a hierarchical

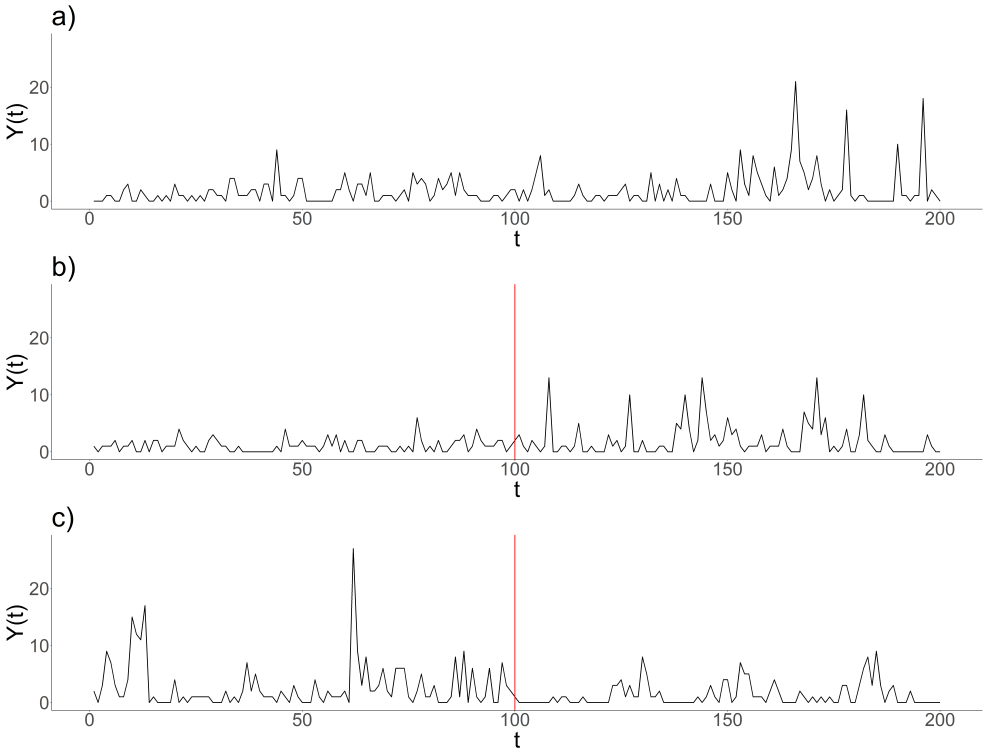


Figure 4: Simulated time series of count with latent AR(1) following (3). The latent time series are displayed in Figure 3. a) No breaks $\rho = 0.5, \sigma^2 = 1$. b) change at $T = 100$ in variance $\sigma_1 = 0.25$, $\sigma_2 = 1$, $\rho = 0.5$. c) change at $T = 100$ in correlation $\rho_1 = 0.5$, $\rho_2 = 0.8$, $\sigma = 1$.

structure and its probability structure is studied closer in the following section.

2.2 Latent Gaussian count model

Given a latent time series $\mathbf{x}$ with parameters $\boldsymbol{\theta}$ the count process $\mathbf{y}$ would have the following conditional probability density function,

$$\pi(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{n} e^{-\lambda_t} \lambda_t^{y_t} / y_t!, \tag{4}$$

by knowing what the latent time series is one knows each Y_i up to some Poisson distribution. Probability density functions (pdf) are denoted using π. For instance $\pi(\mathbf{y}|\mathbf{x})$ denotes the point density function of the counts $\mathbf{y} = (y_1, \ldots, y_n)^T$ given the latent AR(1) time series $\mathbf{x} = (x_1, \ldots, x_n)^T$.

The variance and correlation in every i-th segment are assumed to follow some priors $\pi(\sigma_i^2)$ and $\pi(\rho_i)$, respectively, and independence is assumed between the two. So with m as the number of breaks, $\pi(\boldsymbol{\theta}) = \prod_{i=1}^{m} \pi(\rho_i)\pi(\sigma_i^2)$ is the

prior of $\boldsymbol{\theta}$. Choice of priors will be briefly discussed in Section 2.4. From Bayes' theorem and the law of total probability one has the following,

$$\pi(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}) = \frac{\pi(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})\pi(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int \pi(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})\pi(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d(\mathbf{x}, \boldsymbol{\theta})} = k\pi(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})\pi(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}), \quad (5)$$

where $k \in \mathbb{R}$ is a normalizing constant.

We now specify each of the probability densities above, firstly without breaks,

$$\begin{aligned}\pi(\mathbf{x}|\sigma^2, \rho) &= \phi(x_1; 0, \sigma^2(1-\rho^2)^{-1}) \prod_{t=2}^{n} \phi(x_t; \rho x_{t-1}, \sigma^2) \\ &= (2\pi)^{-n/2}\sigma^{-n}(1-\rho^2)^{1/2} \exp\left[-\frac{1}{2\sigma^2}\left((1-\rho^2)x_1^2 + \sum_{t=2}^{n}(x_t - \rho x_{t-1})^2\right)\right] \\ &= (2\pi)^{-n/2}\sigma^{-n}(1-\rho^2)^{1/2} \exp\left[-\frac{1}{2}\mathbf{x}^T\mathbf{Q}(\rho, \sigma^2)\mathbf{x}\right].\end{aligned} \quad (6)$$

$\phi(\cdot; \mu, \sigma^2)$ denotes the point density function of the normal distribution with mean μ and variance σ^2, the precision (inverse covariance) matrix is,

$$\mathbf{Q}(\rho, \sigma^2) = \frac{1}{\sigma^2}\begin{bmatrix} 1 & -\rho & & & \\ -\rho & 1+\rho^2 & \ddots & & \\ & \ddots & \ddots & & \\ & & & 1+\rho^2 & -\rho \\ & & & -\rho & 1 \end{bmatrix}. \quad (7)$$

Now introduce a breakpoint $1 < T < n$ into the latent model and change the latent model into 2. This gives,

$$\pi_T(\mathbf{x}|\rho_1, \rho_2, \sigma_1^2, \sigma_2^2) = \phi(x_1; 0, \sigma_1^2(1-\rho_1^2)^{-1}) \prod_{t=2}^{T} \phi(x_t; \rho_1 x_{t-1}, \sigma_1^2) \prod_{t=T+1}^{n} \phi(x_t; \rho_2 x_{t-1}, \sigma_2^2)$$

$$\begin{aligned}\pi_T(\mathbf{x}|\rho_1, \rho_2, \sigma_1^2, \sigma_2^2) =& (2\pi)^{-n/2}\sigma_1^{-T}\sigma_2^{-(n-T-1)}(1-\rho_1^2)^{1/2} \\ & \cdot \exp\left[-\frac{1}{2}\left(\sigma_1^{-2}(1-\rho_1^2)x_1^2 + \sigma_1^{-2}\sum_{t=2}^{T}(x_t - \rho_1 x_{t-1})^2 + \sigma_2^{-2}\sum_{t=T+1}^{n}(x_t - \rho_2 x_{t-1})^2\right)\right]\end{aligned} \quad (8)$$

$$\pi_T(\mathbf{x}|\rho_1, \rho_2, \sigma_1^2, \sigma_2^2) = (2\pi)^{-n/2}\sigma_1^{-T}\sigma_2^{-(n-T-1)}(1-\rho_1^2)^{1/2} \exp\left[-\frac{1}{2}\mathbf{x}^T\mathbf{Q}(\rho_1, \rho_2, \sigma_1^2, \sigma_2^2)\mathbf{x}\right] \quad (9)$$

$$\mathbf{Q}(\rho_1, \rho_2, \sigma_1^2, \sigma_2^2) = \begin{bmatrix} \mathbf{Q}_1(\rho_1, \rho_2, \sigma_1^2, \sigma_2^2) & \mathbf{A} \\ \mathbf{A}^T & \mathbf{Q}_2(\rho_1, \rho_2, \sigma_1^2, \sigma_2^2) \end{bmatrix} \quad (10)$$

$$\mathbf{Q}_1(\rho_1, \rho_2, \sigma_1^2, \sigma_2^2) = \sigma_1^{-2}\begin{bmatrix} 1 & -\rho_1 & & & \\ -\rho_1 & 1+\rho_1^2 & \ddots & & \\ & \ddots & \ddots & & \\ & & & 1+\rho_1^2 & -\rho_1 \\ & & & -\rho_1 & 1+\sigma_1^2\sigma_2^{-2}\rho_2^2 \end{bmatrix} \quad (11)$$

$$\mathbf{Q}_2(\rho_1, \rho_2, \sigma_1^2, \sigma_2^2) = \sigma_2^{-2} \begin{bmatrix} 1+\rho_2^2 & -\rho_2 & & & \\ -\rho_2 & 1+\rho_2^2 & \ddots & & \\ & \ddots & \ddots & & \\ & & & 1+\rho_2^2 & -\rho_2 \\ & & & -\rho_2 & 1 \end{bmatrix} \tag{12}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -\rho_2 & \mathbf{0} \end{bmatrix}. \tag{13}$$

$\mathbf{Q}$ is $(n \times n)$, $\mathbf{Q}_1$ is $(T \times T)$ and $\mathbf{Q}_2$ is $(n-T) \times (n-T)$, while $\mathbf{A}$ is a $T \times (n-T)$ matrix. For a situation with more change points we can easily extend this model by introducing parameters $\sigma_3^2, \ldots,$ and $\rho_3, \ldots$ in a similar fashion. Note that both (6) and (9) are multivariate Gaussian centered around $\mathbf{0}$ with variance $\mathbf{Q}^{-1}$. $\mathbf{Q}$ is in both cases a sparse band matrix, because of the properties of AR(1) models. With a traditional time series as response instead of a count values (4) would change to the probability of the product of independent Gaussian variables.

An option is to also allow for large jumps in the value of x_T at a break. This can e.g. be done by letting $x_t = \mu_i + \rho_i x_{t-1} + \epsilon_{it}$, where one assumes μ_i to follow some Gaussian distribution. We, however, choose to focus on variance and correlation changes here which are usually more difficult to detect, and considered to be nuisance parameters in the statistical model. Another potential issue is that in cases with high correlation the time series can often seem jump a consequence of a break in the value of μ.

2.3 INLA

A test for deciding whether or not to introduce a split could be based on the marginals of the hyperparameters. This text will base hypothesis testing on just that. We will either decide on splitting based on overall metrics of quality of fit or base decisions on comparing posterior marginals as $\pi(\boldsymbol{\theta}|\mathbf{y})$. To calculate these we use the INLA scheme. A brief explanation on how INLA approximates the posteriors, based on Rue et al. (2009),Martino and Riebler (2019), the homepage of the INLA project (www.r-inla.org), and the documentation of the R library will now be given.

Instead of simulating data the INLA scheme uses numerical approximation. INLA builds on latent Gaussian models, where given some latent gaussian model one assumes conditional independence of the response. The model described in (3) is such a model, conditioned on the latent time series the response is independent. As a consequence of (6) and (9) and from the fact that $\mathbf{Q}$ is sparse, the model a perfect case for INLA, as model fitting is likely to be fast. For models without breakpoint denote the hyperparameters with $\boldsymbol{\theta} = (\rho, \sigma^2)$ and similarly $\boldsymbol{\theta} = (\rho_1, \rho_2, \sigma_1^2, \sigma_2^2)$ for models with break.

One aim of INLA is to find accurate descriptions of the hyperparameter posteriors, in the case when (1) is latent that would be $\pi(\rho|\mathbf{y})$ and $\pi(\sigma^2|\mathbf{y})$. Another goal of the scheme is to approximate the marginal posteriors of the hyperparameters $\boldsymbol{\theta}$ and the marginal posteriors of $\mathbf{x}$. That will say to approximate the following integrals,

$$\pi(x_i|\mathbf{y}) = \int \pi(x_i|\boldsymbol{\theta}, \mathbf{y})\pi(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}, \tag{14}$$

$$\pi(\theta_j|\mathbf{y}) = \int \pi(\boldsymbol{\theta}|\mathbf{y}) d\boldsymbol{\theta}_{-j}, \tag{15}$$

where $\boldsymbol{\theta}_{-j}$ is the vector $\boldsymbol{\theta}$ with the j-th component removed.

To numerical approximate the posterior marginals each component in the integrand needs to be approximated. We explain how the scheme goes forth in doing so. Depending on whether or not there is a breakpoint one can insert (6) or (9) to get,

$$\pi(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}) \propto \pi(\boldsymbol{\theta})|\mathbf{Q}(\boldsymbol{\theta})|^{n/2} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}(\boldsymbol{\theta})\mathbf{x} + \sum_{i=1}^{n} \log \pi(y_i|x_i, \boldsymbol{\theta})\right). \tag{16}$$

This can in turn be used to find parts of the integrands in (14) and (15). The following approximation is used in creating the joint marginal posterior of the hyperparamters,

$$\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y}) \propto \left.\frac{\pi(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta})}{\tilde{\pi}_G(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})}\right|_{\mathbf{x}=\mathbf{x}^*(\boldsymbol{\theta})}. \tag{17}$$

$\tilde{\pi}_G(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$ is a Gaussian approximation to $\pi(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$ built by matching the mode and curvature at the mode $\mathbf{x}^*(\boldsymbol{\theta})$. In short, $\tilde{\pi}_G(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$ can be looked upon as a Taylor approximation to the second order around the mode, and is equivalent to the Laplace approximation. With the above expression, and some more, explained in the aforementioned papers finding (15) is achievable.

Next the scheme creates an approximation of $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$ by using the same trick,

$$\pi(x_i|\boldsymbol{\theta}, \mathbf{y}) \propto \frac{\pi(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})\pi(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\pi(\mathbf{x}_{-i}|x_i, \boldsymbol{\theta}, \mathbf{y})} \approx \left.\frac{\pi(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})\pi(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\tilde{\pi}_G(\mathbf{x}_{-i}|x_i, \boldsymbol{\theta}, \mathbf{y})}\right|_{\mathbf{x}_{-i}=\mathbf{x}^*_{-i}(x_i,\boldsymbol{\theta})} = \tilde{\pi}(x_i|\boldsymbol{\theta}, \mathbf{y}), \tag{18}$$

Laplace approximation is used at the mode to estimate the denominator, now with x_i fixed. The idea is then to investigate $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$ at grid of $\boldsymbol{\theta}$ and create the approximations,

$$\tilde{\pi}(x_i|\mathbf{y}) = \sum_k \tilde{\pi}(x_i|\boldsymbol{\theta}^{(k)}, \mathbf{y})\tilde{\pi}(\boldsymbol{\theta}^{(k)}|\mathbf{y})\Delta\boldsymbol{\theta}^{(k)}, \tag{19}$$

where $\boldsymbol{\theta}^{(k)}$ is part of selected support points, $\Delta\boldsymbol{\theta}^{(k)}$ being the distance in between points. With the R INLA package implementing the above becomes a relatively easy task.

2.4 Choosing priors

We base our discussion of priors on Simpson et al. (2015) and the INLA documentation. For the models fitted in this text INLA's default priors for the AR(1) latent time series which are types of Penalized Complexity (PC) priors are used. PC priors are used due to their flexibility. They have the form $\pi_{\boldsymbol{\xi}}(\boldsymbol{\theta})$ where $\boldsymbol{\xi}$ alter the complexity of the prior, changing the parameters for instance allows a reduction of over-fitting.

The prior that is used for precision of the time series is on the following form,

$$\pi(\sigma^2) = \frac{\lambda}{2\sigma}\exp(-\lambda\sigma) \tag{20}$$

$$\lambda = -\frac{\ln(\alpha)}{u}. \tag{21}$$

The parameters of the model are u and α and its defaults are $u = 1$, $\alpha = 5e-5$. The idea is that the parameters alter the following probability,

$$P(\sigma > u) = \alpha. \tag{22}$$

For instance, increasing α keeping u constant increase the chance of a higher variance in the latent time series.

The prior for ρ is,

$$\pi(\rho) = \lambda \exp(-\lambda\psi(\rho))J(\rho), \tag{23}$$

where,

$$\psi(\rho) = (-\log(1-\rho^2))^{-1/2} \tag{24}$$

$$J(\rho) = \frac{|\rho|}{\psi(\rho)(1-\rho^2)} \tag{25}$$

$$\lambda = -\log(\alpha)/\psi(u). \tag{26}$$

By default parameters $u = 0, \alpha = 0.15$ are used. The idea is again to use the prior and the parameters to alter the probability of large values of the hyperparameter. In this case the idea is that the parameters of the PC prior alter,

$$P(|\rho| > u) = \alpha. \tag{27}$$

3 Methods for change point detection

This text takes two approaches in structural break detection, the first is when the quality of model fit varies, described in Section 3.1 and 3.2. The second approach is to classify a break as to when the posterior marginals of a given parameter over a break differs significantly explained more in depth in Section 3.3.

Now that the model which will be used has been specified, we now how breakpoints will be identified. It is required that each split node contains at least 50 observation points. As a consequence, if one has a time series of 150 points one can only find breakpoints on observation point 50 to 100. This is to ensure that the metrics has some chance on converging to their true values. It will be assumed that breaks only happen in-between observation point, if one has observations of trade volume over a span of 150 days, a break can i.e. happen at the following days:

$$50.5, 51.5, 62.5, \ldots 99.5. \tag{28}$$

The method used is based on Algorithm 1. In the algorithm, the main idea is to evaluate models at different break points. One calculates metrics of model fit at possible split points and compare the models with breaks to one without. It is then evaluated whether or not to introduce a split found at the optimal split point. To avoid fitting models at all possible split points smoothed splines are used to create a estimation the value of the metric against the split points that have not been evaluated. In Algorithm 1, k is the number of initial fits and m

Algorithm 1 Find most likely break point

1: Fit initial model $\mathcal{M}_0$ assuming no break point.
2: Fit initial models $\mathcal{M}_{11}, \mathcal{M}_{1k}$ assuming breaks at $p_1, \dots p_k$ respectively.
3: With some metric calculated using a function, $f : \mathcal{M}_0, \mathcal{M}_{1i} \rightarrow d$, fit a smoothing spline.
4: **for** i in $1, ..., m$ **do**
5: Fit a new model $\mathcal{M}_{1(k+i)}$ At the best possible not searched point p_{k+i} estimated from spline.
6: Update the smoothed spline.
7: **end for**
8: Return the found model for the best split point.

is the number of spline model fits done. In the simulations in the later section $k = 50$ split points are initially calculated and used to create a smoothed spline. One then fit and calculate the metric value at the estimated maximum of the spline. This is repeated $m = 25$ times. The point with the optimal metric value that favor splitting will be deemed as the best split point.

A brief summary on the theory behind smoothing splines based on James et al. (2013) is given. Assume one has observed pairs of (x_i, y_i) and wants to find a $g(x)$ such that $g(x_i) \approx y_i$ is as good as possible smooth approximation. Choose to find $g(x)$ by minimizing,

$$\sum (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt, \tag{29}$$

where $\lambda > 0$ is a tuning paramater. James et al. (2013) gives that this is equivalent to fitting a natural cubic spline with knots at each observation point. An example of how a spline if fitted is displayed in Figure 5. When hypothesis

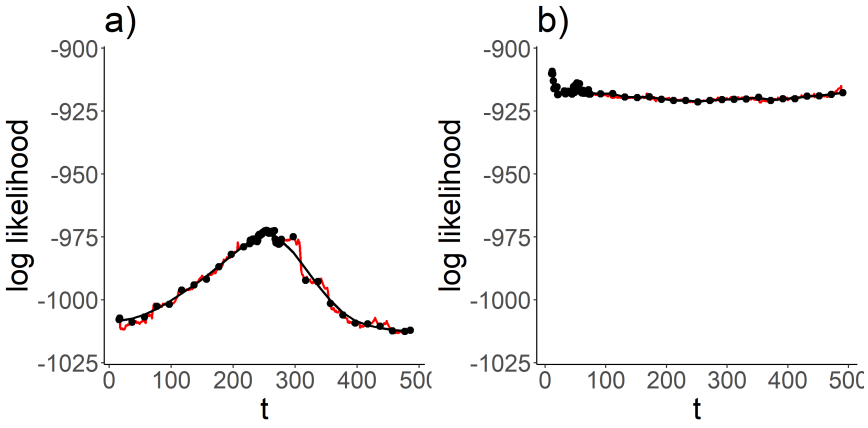


Figure 5: Spline fit (black) against fitting at all split points (red), dots are fitted models in the spline case. a) Case with break in correlation with T = 250. $\rho_1 = 0.5, \rho_2 = -0.5, \sigma = 1$. b) Case without break $\rho_1 = 0.5, \sigma = 1$.

$\mathcal{H}_0$ is true the parameters stable across the whole predictor space and thus

introducing a breakpoints should at least asymptotically have no effect on the metric as the estimated parameters should be the same. This is reflected in Figure 5b. If hypothesis $\mathcal{H}_1$ is true likelihood at the correct split point would be higher than without one, and the likelihood would have a maxima similar to that of a) in Figure 5. The metric seems to be good at identifying splits, at least in the simulated case displayed in the figure.

In practice decision boundaries for the different metrics when deciding on a split will be simulated. Algorithm 2 describes how this threshold is found in the real data application. In the real data application we use $k = 25$.

Algorithm 2 Simulate a decision boundary.

1: Fit a model $\mathcal{M}_0$ assuming there is no break point using observation data $\mathbf{y}$.
2: Fit a model $\mathcal{M}_1$ at optimal break point using observation data $\mathbf{y}$ as described in Algorithm 1.
3: Using found model parameters in $\mathcal{M}_0^*$ simulate data $\mathbf{y}_1^*, \dots \mathbf{y}_k^*$ assuming that there is no break point.
4: Fit models $\mathcal{M}_{01}^*, \dots \mathcal{M}_{0k}^*$ on $\mathbf{y}_1^*, \dots \mathbf{y}_k^*$ assuming no break.
5: Fit models $\mathcal{M}_{11}^*, \dots \mathcal{M}_{1k}^*$ on $\mathbf{y}_1^*, \dots \mathbf{y}_k^*$ assuming break at same points as $\mathcal{M}_1$.
6: Using model pairs $(\mathcal{M}_{01}^*, \mathcal{M}_{11}^*), \dots$ calculate metrics $d_1^*, \dots d_k^*$ and use these to estimate a confidence interval for the metric.
7: Using model pair $(\mathcal{M}_0, \mathcal{M}_1)$ calculate metric d.
8: If d is outside of the confidence interval decide on splitting, else decide on not introducing a split.

3.1 Model comparison using marginal likelihood

The main metric we use to compare models is the marginal likelihood. It is commonly used in comparing Bayesian models and with the INLA-package evaluating the marginal model likelihood is also quite fast. Hubin and Storvik (2016) discuss the use and calculation of marginal likelihoods in the INLA-package, but in short INLA estimates the marginal likelihood by approximating the following,

$$p(\mathbf{y}) \approx \int \left. \frac{\pi(\mathbf{y}, \boldsymbol{\theta}, \mathbf{x})}{\tilde{\pi}_G(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})} \right|_{\mathbf{x}=\mathbf{x}^*(\boldsymbol{\theta})} d\boldsymbol{\theta}, \tag{30}$$

the integrand here is made up of parts that are already calculated by the framework. Furthermore, as $p(\mathbf{y})$ is the normalizing constant in (17), there is little additional work that needs to be done to evaluate the marginal likelihood.

When comparing models using marginal log likelihood we use the following statistic:

$$LR = \log p_1(\mathbf{y}) - \log p_0(\mathbf{y}), \tag{31}$$

where $p_0(\mathbf{y})$ is the marginal log likelihood of the model assuming that there is no break, and $p_1(\mathbf{y})$ is the marginal log likeliood of the model assuming there is a break. A high value of LR would mean that the model with break is more likely, and that we should consider including a breakpoint. Figure 5 displays realizations of splines fitted with marginal log likelihoods.

3.2 Splitting by using the deviance information criteria

Another common metric to used when comparing Bayesian models is the Deviance information criteria (DIC). Chen and Lee (2016) for instance uses this approach when evaluating a split. The DIC is a measure of complexity and fit, and can be written as,

$$DIC = \bar{D} + p_D. \tag{32}$$

The metric is presented in Spiegelhalter et al. (2002). $\bar{D}$ is the posterior mean of the deviance, and p_D is the effective number of parameters. So smaller values of DIC is preferred. With INLA it is easy to produce the DIC metric as the DIC is a few calculations a way from the calculations in the INLA scheme. In our simulation study we test how well DIC functions as a splitting criteria.

When comparing models using DIC we use the following statistic,

$$d = DIC_1 - DIC_0, \tag{33}$$

where DIC_0 is the DIC of the model assuming that there is no break, and DIC_1 is the DIC of the the model assuming there is a break. An idea is to accept a split when $d < 0$, similarly to what is done in Chen and Lee (2016). However, we experienced that doing this gave relatively unstable results, and we thus decide on simulating the decision boundary.

3.3 Splitting based on marginal posteriors

As the marginal posterior densities of the hyperparameters are found, an idea could be to compare the posterior densities of the different nodes. A thought would be to use Kullback Leibler divergence as a metric for difference, it is described in Kullback (1968) and can be written as,

$$kbl(p(\boldsymbol{\theta}), q(\boldsymbol{\theta})) = \int p(\boldsymbol{\theta}) \log\left(\frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}\right) d\boldsymbol{\theta}. \tag{34}$$

$p(\boldsymbol{\theta})$, and $q(\boldsymbol{\theta})$ would be the marginal posteriors of the hyperparameter $\boldsymbol{\theta}$ that we want to compare at different nodes. A large value would mean large difference between the two. However, in our case these point densities often take the value of something close to 0, making the aforementioned metric difficult to handle.

As a consequence we choose to use the L2 norm as an alternative in comparing the marginal posteriors. Using,

$$d = ||p(\boldsymbol{\theta}) - q(\boldsymbol{\theta})||_2^2 = \int \left(p(\boldsymbol{\theta}) - q(\boldsymbol{\theta})\right)^2 d\boldsymbol{\theta} \tag{35}$$

as a metric. In both cases decide on whether or not a split is made when the statistic above is over a given threshold, in practice this threshold is found by simulation. It is expected that this metric will perform relatively well in cases with large differences. However, it is likely that it will struggle in cases where there are small differences between the parameters.

4 Simulation Study

We now want to test how well our method fares, and want to compare different decision metrics that can be used. In section 4.1 we check the model behaviour

for a single simulation case using marginal log likelihoods as a split metric. In section 4.2 and in section 4.3 we check how well the methods presented earlier perform at finding breakpoints.

4.1 Breakpoint detection on simulated data

In Figure 6 breakpoint detection with the method described above is applied on simulated data, marginal likelihood is used as a criteria for splitting and the best change point is accepted without any further testing. The simulated case is with a break in correlation and the following parameters are used $\rho_1 = 0.5, \rho_2 = 0.8, \sigma = 1, T = 100$. The latent AR(1) is the realization displayed in Figure 4c. The estimated 95% credible intervals are displayed both for the latent AR(1) and for the response. The model fit seems to fit the true data well, the true latent AR(1) is mostly within the 95% credible band and the found split is close to the true split value. Furthermore the marginals of the hyperparameters seem to be centered around their true values, which indicated a good model fit. The model does not seem to over fit the data, so the complexity of the PC prior does not need to be changed.

4.2 Finding structural change in correlation with constant variance

Assume now a time series with count data is given. The null hypohtesis on the data is that it follows the setup described in (1), but one want to test if (2) describes the data better, more specifically the hypothesis are, $\mathcal{H}_0$ there is no breakpoint and $\mathcal{H}_1$ there is some breakpoint $1 < T < n$ where correlation changes. We want to see how well the methods described in Section 3 performs. We simulate data for both when $\mathcal{H}_0$ is true and for when $\mathcal{H}_1$ is true. We use the results from $\mathcal{H}_0$ to estimate a decision boundary for the different metrics, and test how it performs in detecting splits when applied to data simulated when $\mathcal{H}_1$ is true. The numbers of count observations are $n = 150, 300, 500, 1000$. When $\mathcal{H}_1$ is true we set the break-point to be $\lfloor n/2 + u \rceil$ where $u \sim Uniform(0, 10)$. We run 500 simulations assuming $\mathcal{H}_0$ to be true, and 500 cases assuming $\mathcal{H}_1$ is true. We evaluate how decision metrics based on the 0.95 quantile of the $\mathcal{H}_0$ data (0.05 when using DIC) would fare when applied to the $\mathcal{H}_1$. We then look at the Type II error. We calculate the LR difference, DIC difference, and the L2 norm difference of posterior marginals of the correlation hyperparameters and use them as metrics to detect breakpoints. The parameters we simulate are displayed in Table 1 and the results are displayed as estimated density plots for the different metrics, parameters in Figure 7, Figure 8, Figure 9 and Figure 10. Note that the in the display blue represent cases when synthetic data is generated when $\mathcal{H}_0$ is true and green when $\mathcal{H}_1$ is true.

We study the results and first turn to Figure 7 to see how marginal log likelihood performed as a splitting criterion. For the easiest cases (a-d) the method seems to perform quite well. The densities with break and without break differ consistently from each other, indicating that the metric have high enough resolution to spot the cases from each other. For $n = 150$ it seems to have high power, and for $n \geq 300$ it reaches powers close to 1. The same is the case for the medium cases (e-h). However, for the more difficult cases (i-l) power is quite low and does not seem to to increase before one have $n = 1000$

Table 1: Parameters used in simulation, difficulty increases with rows

$\mathcal{H}_0$ true ($\sigma^2 = 1$)	$\mathcal{H}_1$ true ($\sigma^2 = 1$)	
ρ_0	ρ_1	ρ_2
0.75	-0.5	0.75
0.5	-0.5	0.5
0.75	0.5	0.75

observations. Turning to Figure 7 and the DIC the same seems to be the case, the splitting criterion however performs somewhat better at the difficult cases with low observation counts (i-j). The L2 norm, displayed in Figure 9 seems to perform consistently bad in all cases and achieves low levels of power.

We also include a plot that displays the absolute distance in between the mode of the posterior marginals when fitting a model with breakpoint, displayed in Figure 10. For each model fit simulation this effectively is the distance between the modes of the posterior densities equivalent to those displayed in Figure 4.1d). We include this display to ensure that model manages to reflect the true data. From looking at Figure 10 it seems that the models capture the real distance between the hyperparameters. With data generated under $\mathcal{H}_0$ distances are close to 0 and in cases when $\mathcal{H}_1$ is true the distances seem to reflect the real parameter difference. In cases with large difference as the easy and medium case (a-h) the distances are large, while in more difficult cases (i-l) the model struggels differentiating the two hyperparameters.

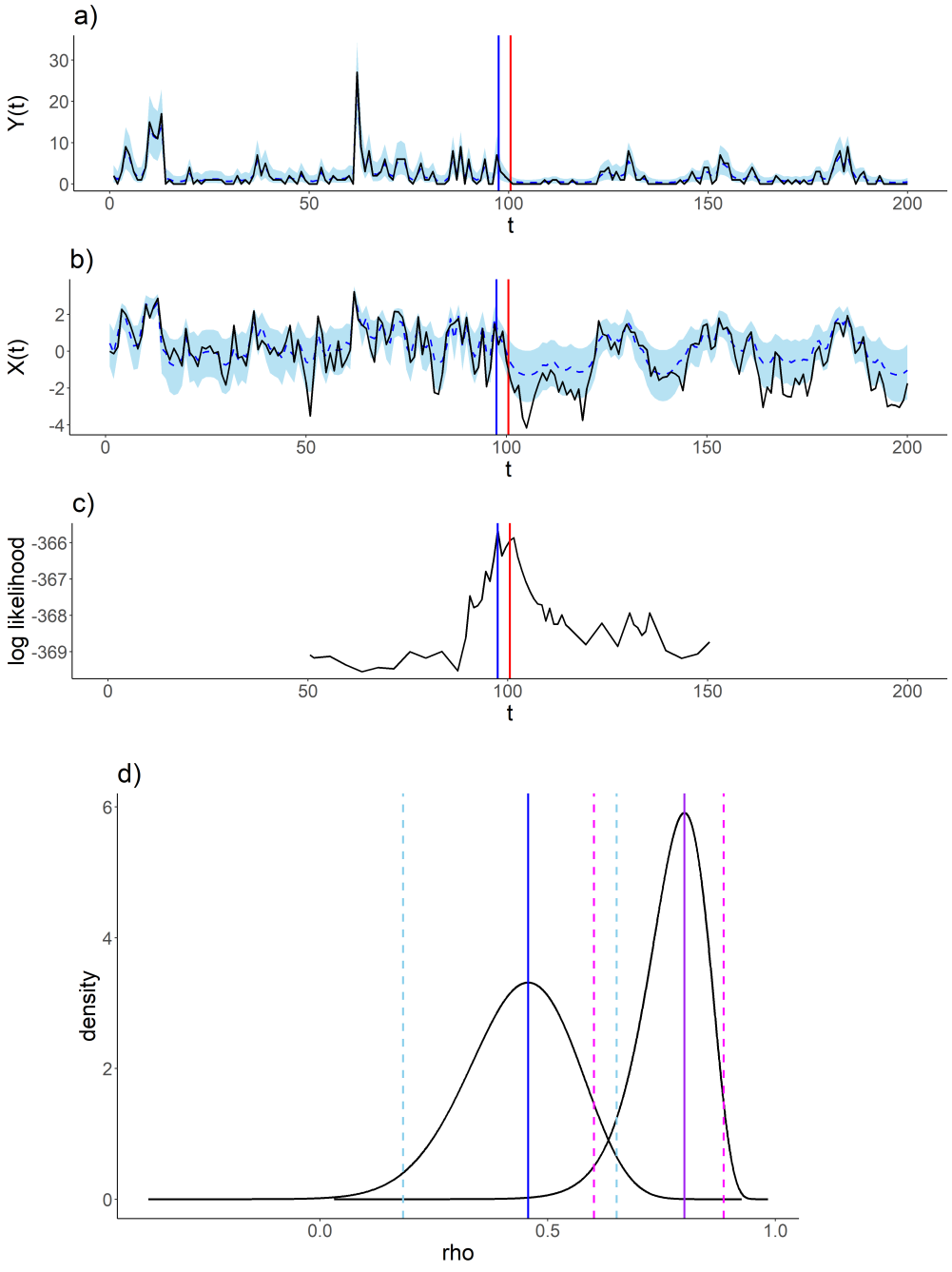


Figure 6: Breakpoint detection with marginal likelihood as split metric. Simulated case with break in correlation, $\rho_1 = 0.5, \rho_2 = 0.8, \sigma = 1, T = 100$. Case is same as Figure 4c. Display with 95% confidence (skyblue), posterior mean (dashed blue) and the true values (black). a) Time series of counts. b) Latent AR(1). c) Marginal log likelihood with split at t. True break displayed as vertical red line, found break is vertical blue line. d) Marginal posterior of the correlation parameter with 95% confidence and mode, left of break is blue, right of break is purple.

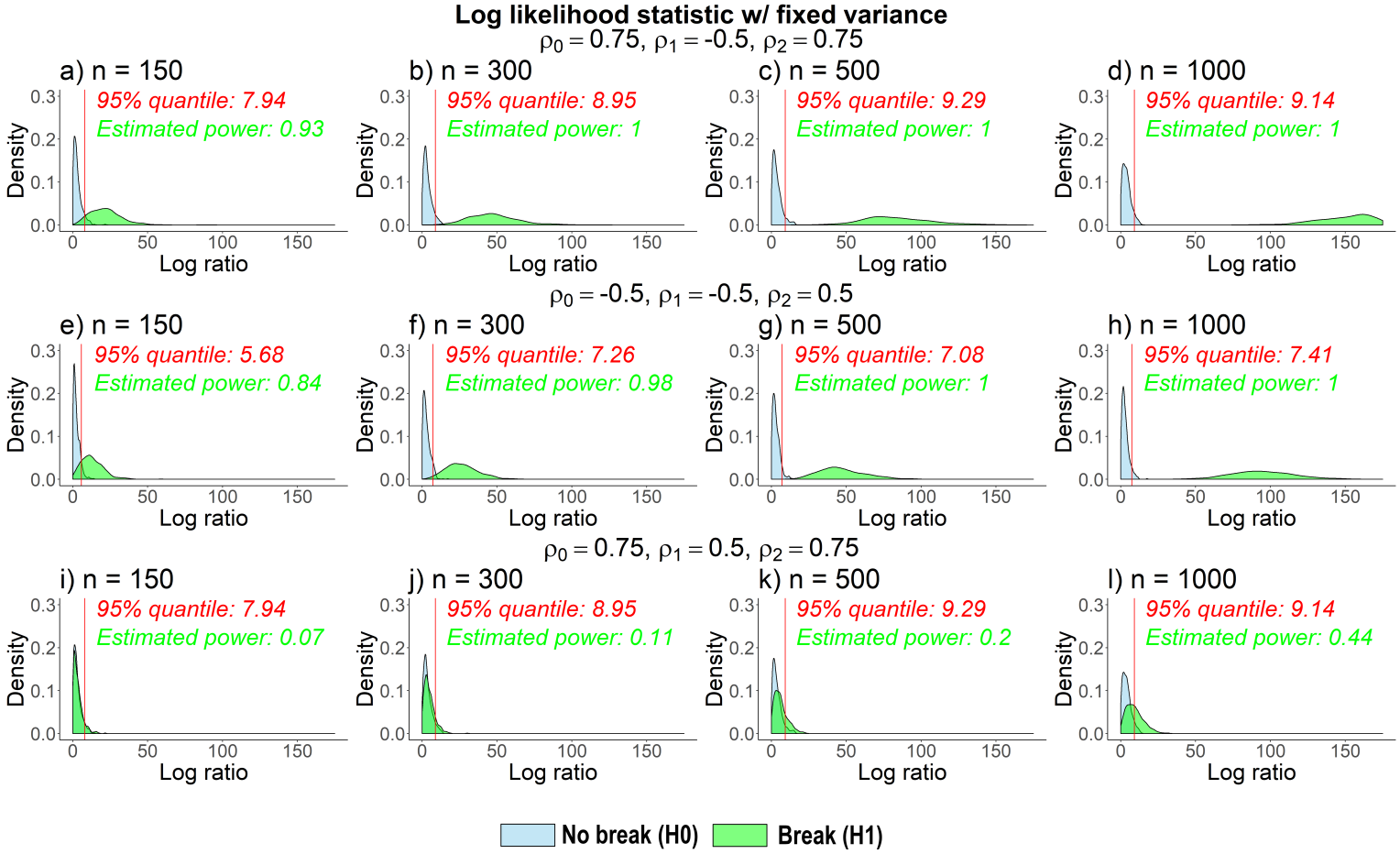


Figure 7: LR of simulations with fixed variance and changing correlation. Each cell display estimation of density of the metrics for simulation with break (green) and without break (blue). Parameters become more difficult to resolve by increasing row. Higher observation count with increased columns. Display 95% quantile of density in cases without break and use it as decision boundary (red vertical line). Applying this boundary on cases with breakpoint the power of the method is estimated.

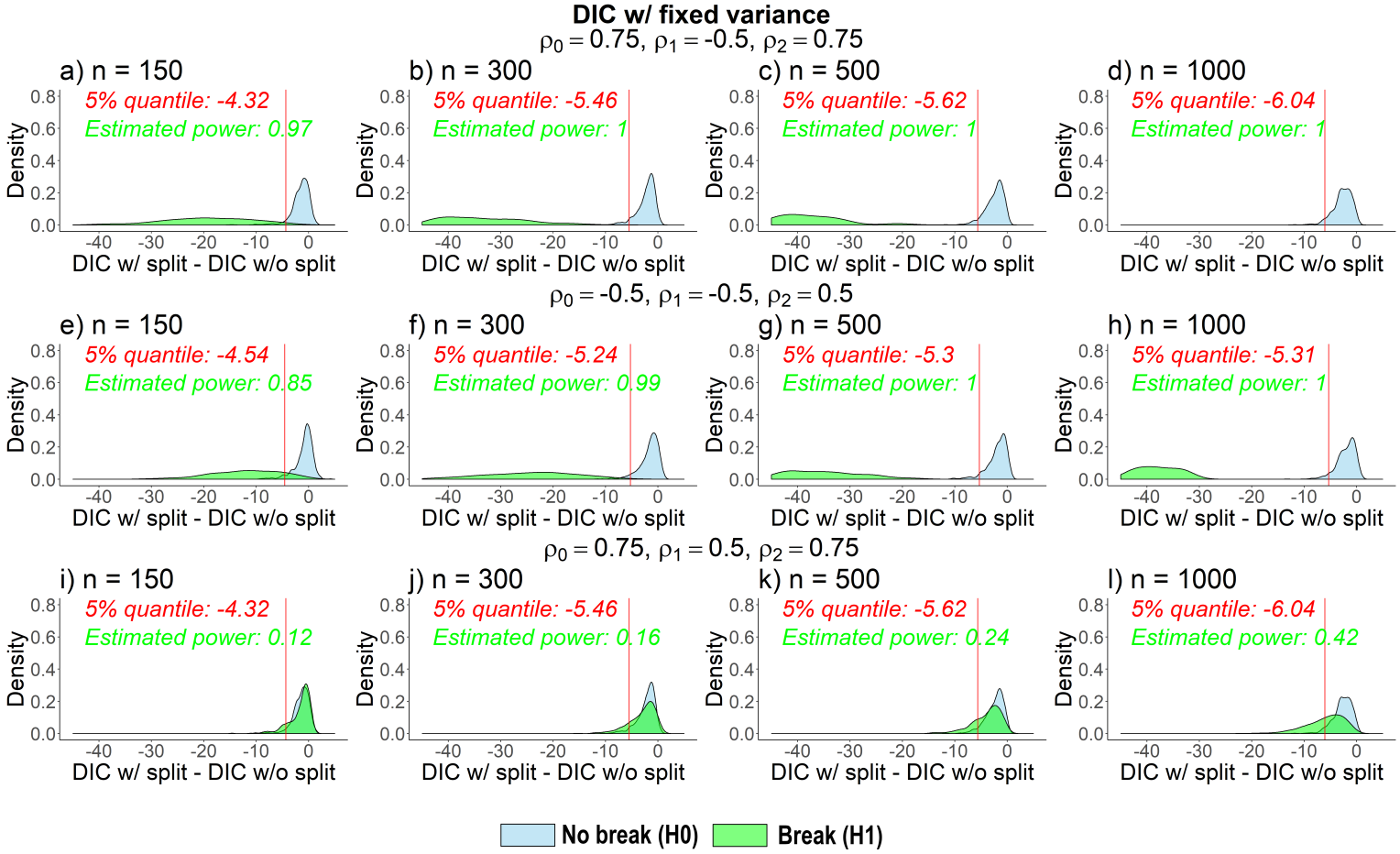


Figure 8: DIC difference of simulations with fixed variance and changing correlation for different parameters. . Each cell display estimation of density of the metrics for simulation with break (green) and without break (blue). Parameters become more difficult to resolve by increasing row. Higher observation count with increased columns. Display 5% quantile of density in cases without break and use it as decission boundary (red vertical line). Applying this boundary on cases with breakpoint the power of the method is estimated.

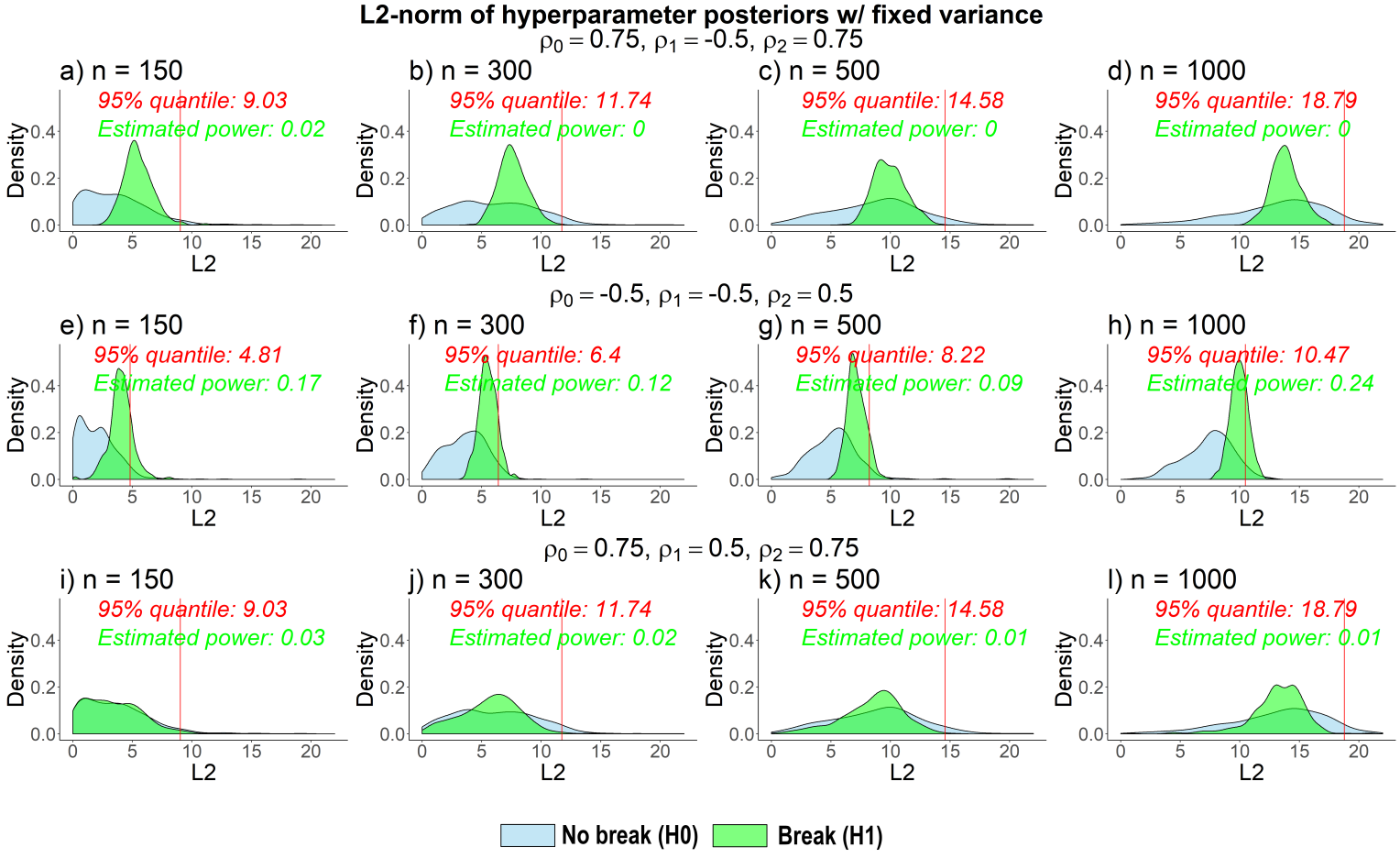


Figure 9: Distribution of L2 distance between marginal posteriors of hyperparameters in simulations with fixed variance and changing correlation. . Each cell display estimation of density of the metrics for simulation with break (green) and without break (blue). Parameters become more difficult to resolve by increasing row. Higher observation count with increased columns. Display 95% quantile of density in cases without break and use it as decission boundary (red vertical line). Applying this boundary on cases with breakpoint the power of the method is estimated.

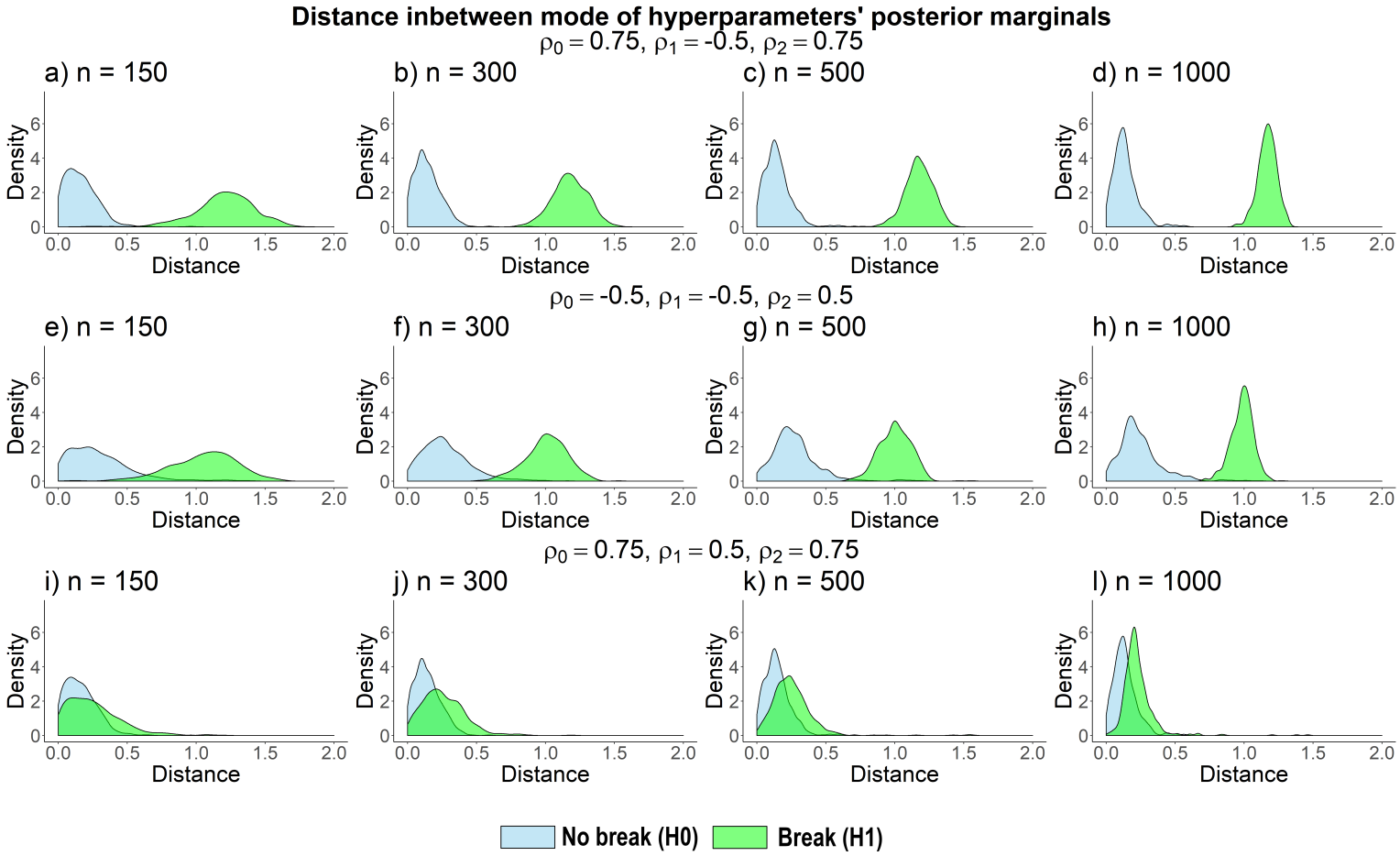


Figure 10: Distribution of distance between mode of marginal posteriors of hyperparameters in simulations with fixed variance and changing correlation.

4.3 Finding structural change in variance with fixed correlation

We now want to test how the different methods presented in Section 3 fare when there is a structural change in variance and fixed correlation. Again, the model follows the setup described in (1), but we want to test if (2) describe the data better. More specifically we again test how well our method is at finding which of the hypothesis $\mathcal{H}_0$ there is no breakpoint and $\mathcal{H}_1$: there is some breakpoint $1 < T < n$ where variance changes is true. The parameters used are displayed in Table 2, other than that the setup is equivalent to the simulations in section 4.2. The results are displayed in Figure 11, Figure 12, Figure 13 and Figure 14.

$\mathcal{H}_0$ true ($\rho = 0.8$)	$\mathcal{H}_1$ true ($\rho = 0.8$)	
σ_0^2	σ_1^2	σ_2^2
0.25	0.25	1
0.5	0.5	1
0.8	0.8	1

Table 2: Parameters used in simulations, becomes increasingly more difficult by increasing row number.

Turning to the results, we first study the display in Figure 11 and the marginal log likelihood. Marginal likelihood seems to performs worse compared to the results in the last section. Power drops in the medium case, and you are dependent on having a lot of observations for it to be high. In the most difficult cases the metric struggles at separating the cases with a break from the cases without, the density plots (k-l) seems to match completely. Turning to DIC and Figure 12 much of the same seems to be the case. However, the metric performs somewhat better in the medium cases (e-h) compared to the marginal likelihood. Lastly we turn to Figure 13 where we used the L2 norm as a metric, compared to the two earlier metrics the L2 achieves higher power levels in nearly all cases.

A plot the absolute distance between the mode of the hyperparameters mode when fitting a model with break point in Figure 14 is also included. There only seem to be a slight difference in the models between the cases when $\mathcal{H}_0$ is true and when $\mathcal{H}_1$ is true for the more difficult cases, explaining the low power levels, as our break point models only seemed to be able to capture the real difference in a few cases.

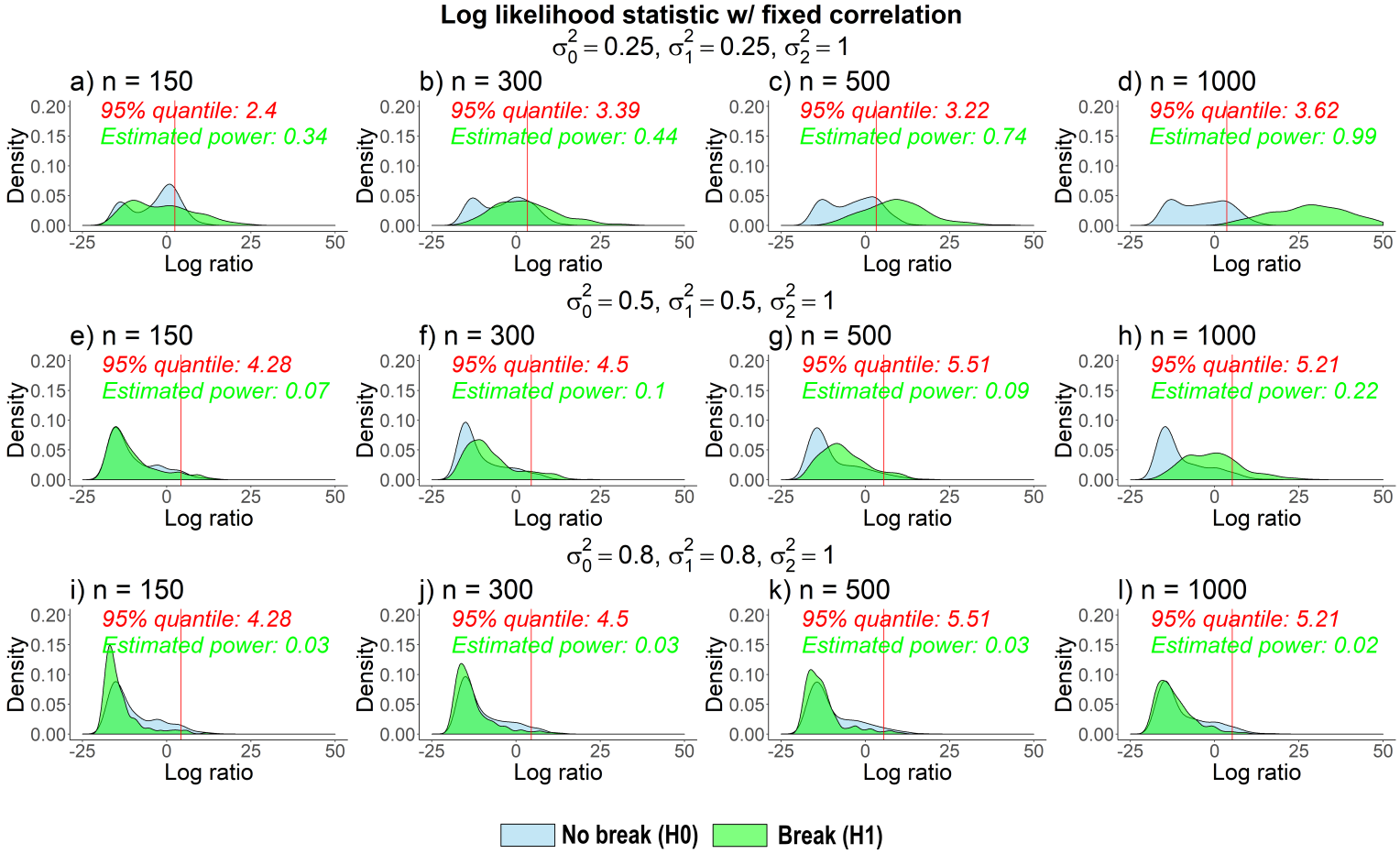


Figure 11: LR of simulations with changing variance and fixed correlation. Each cell display estimation of density of the metrics for simulation with break (green) and without break (blue). Parameters become more difficult to resolve by increasing row. Higher observation count with increased columns. Display 95% quantile of density in cases without break and use it as decision boundary (red vertical line). Applying this boundary on cases with breakpoint the power of the method is estimated.

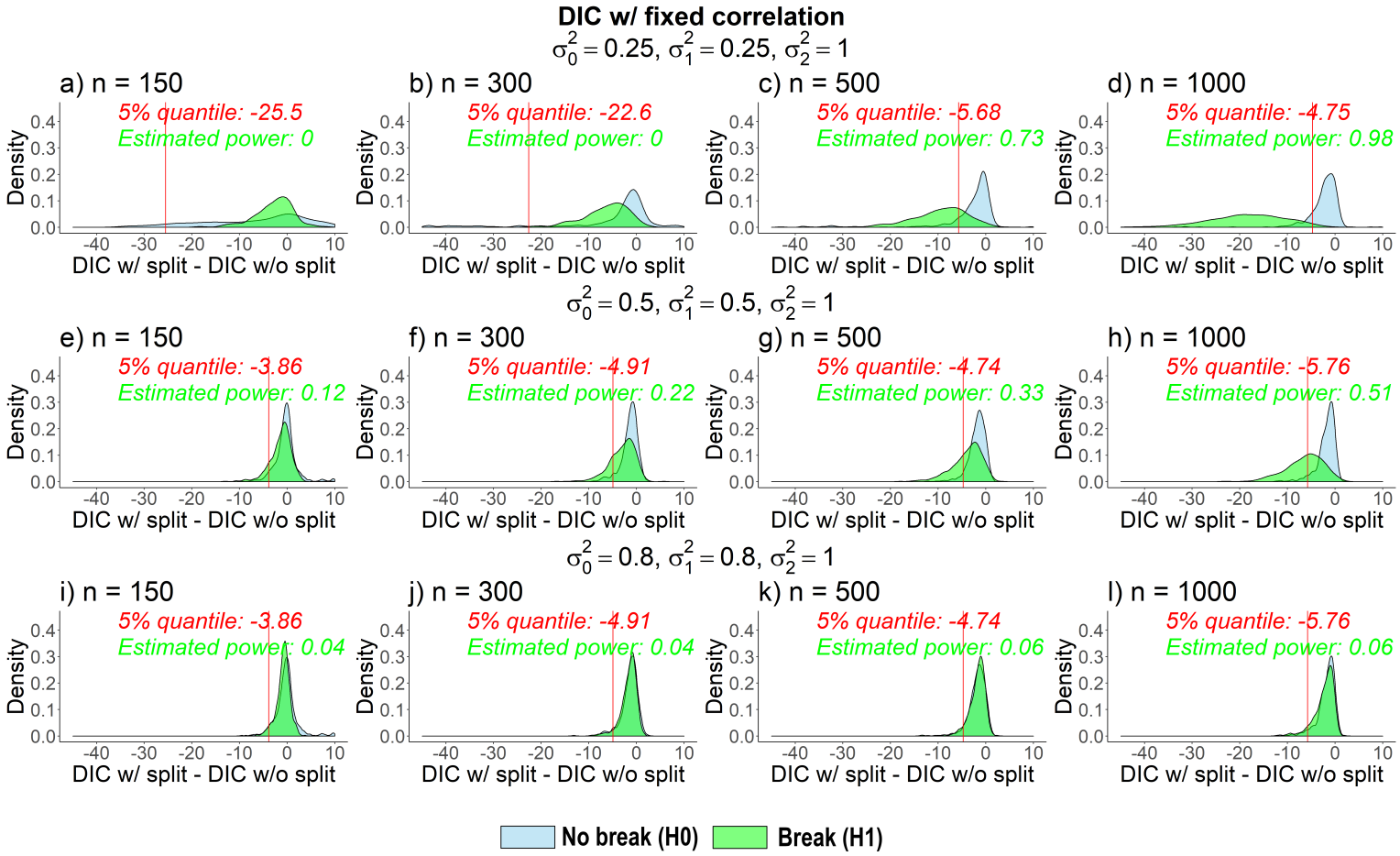


Figure 12: DIC of simulations with changing variance and fixed correlation. Each cell display estimation of density of the metrics for simulation with break (green) and without break (blue). Parameters become more difficult to resolve by increasing row. Higher observation count with increased columns. Display 5% quantile of density in cases without break and use it as decision boundary (red vertical line). Applying this boundary on cases with breakpoint the power of the method is estimated.

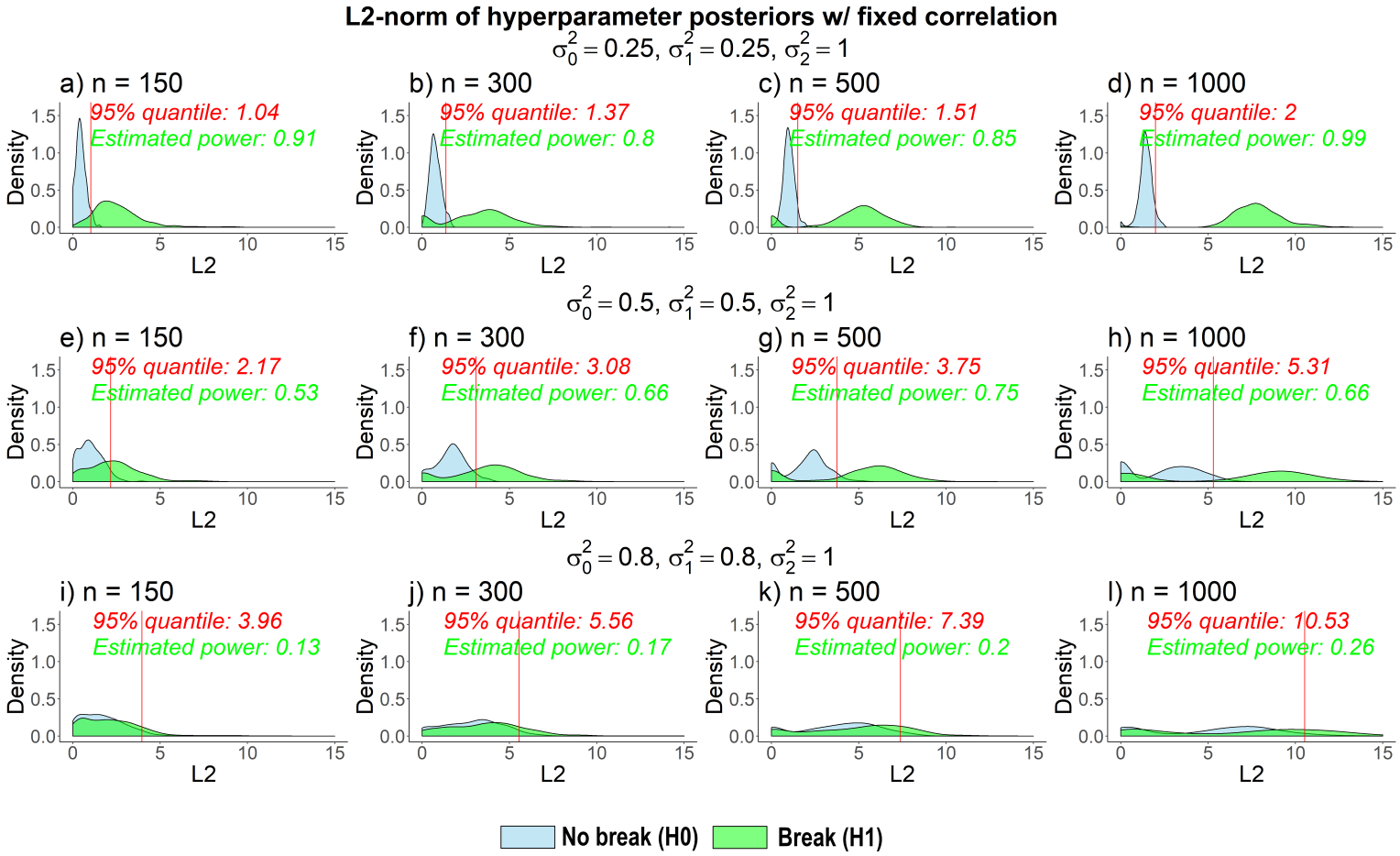


Figure 13: Distribution of F2 distance between marginal posteriors of hyperparameters in simulations with fixed variance and changing correlation. Each cell display estimation of density of the metrics for simulation with break (green) and without break (blue). Parameters become more difficult to resolve by increasing row. Higher observation count with increased columns. Display 95% quantile of density in cases without break and use it as decision boundary (red vertical line). Applying this boundary on cases with breakpoint the power of the method is estimated.

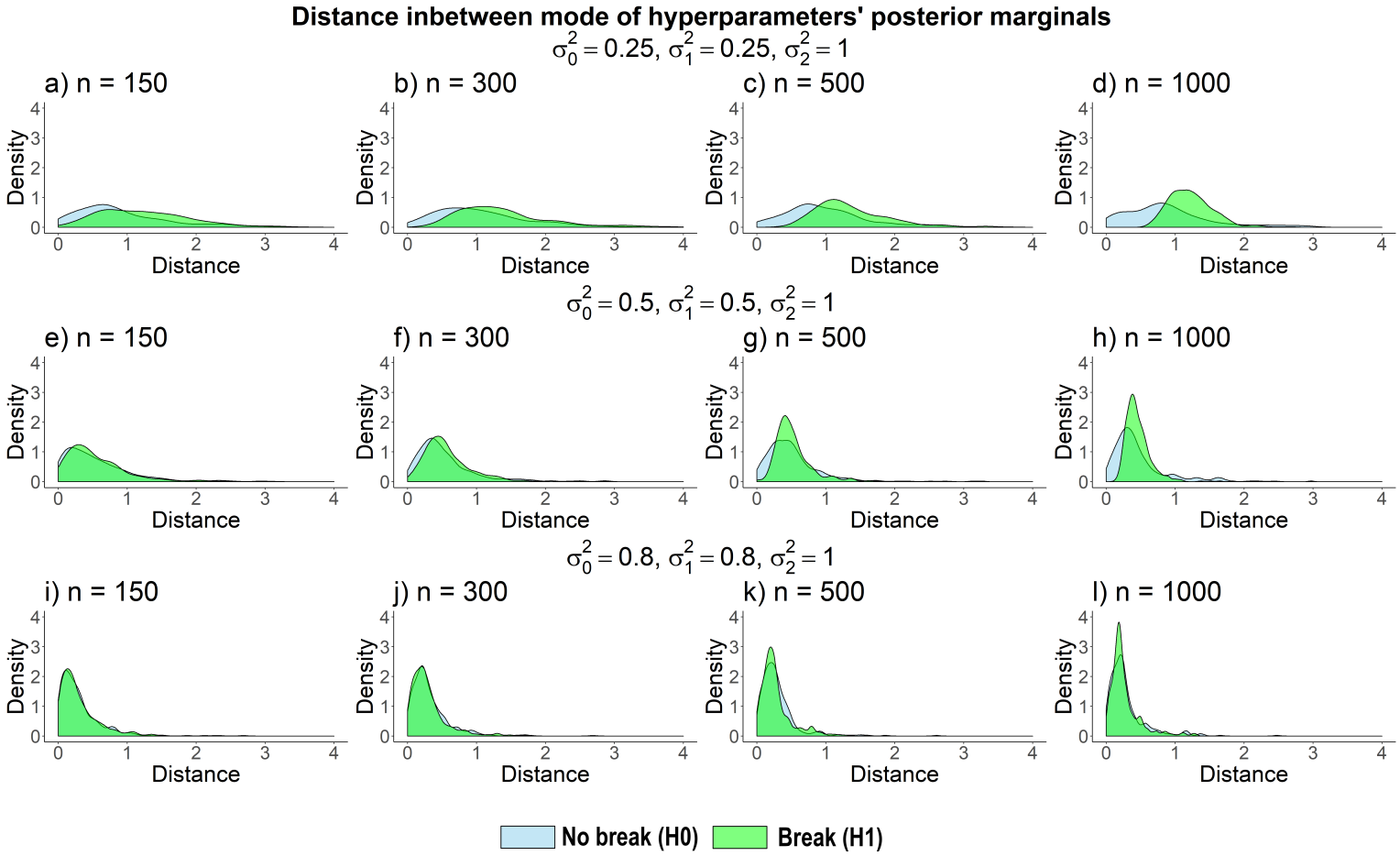


Figure 14: Distribution of distance between mode of marginal posteriors of hyperparameters in simulations with changing variance and fixed correlation.

5 Real data application: Detecting breaks in correlation structure of trading volume

We now turn to real data application, and breakpoint detection in daily trading volume of the TSLA-stock. Trading volume is a measure of how many stocks have been traded in a day. We only look at a segment of the data set, with data after late 2013. The TSLA stock is the stock of the american car producer Tesla and is registered on the NASDAQ stock exchange. The data was retrived from Kaggle. Kaggle is a repository for data sets that can be used in data science and machine learning. The method is applied to check for breaks in the correlation structure in day dependence of the trading volume. A recursive approach is used to detect breaks is implemented, it can be described in two steps,

1. Detect a change point and test for significance using Algorithm 1 and Algorithm 2. Use parameters for initial fit without breakpoints as base for $\mathcal{H}_0$. If not significant at 0.95 significance stop the search.

2. Split the time series in two at found break and start from step 1. using the smaller pieces.

For each partition we do 50 initial break evaluation, this is followed by 25 evaluations where we maximize spline fits based on marginal log likelihood, variance is fixed to what is observed in the initial fit. (In this case $\sigma_0^2 = 1/3.226$). Algorithm 2 is used with 25 repetitions at step 1. and the 95-th quantile is used for deciding whether or not to introduce a split.

Running the above method on the trade volume data yields two significant breakpoints, the change points found are displayed in Figure 15 Considering

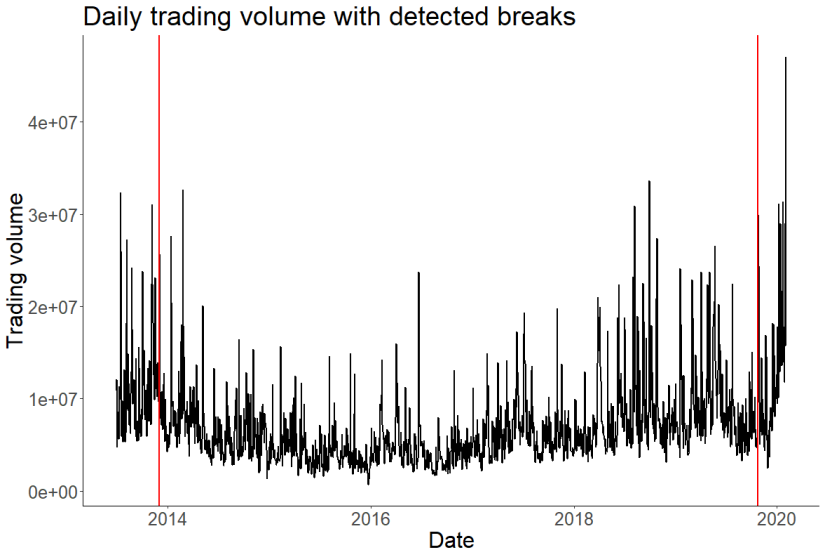


Figure 15: Detected breakpoints in daily trading volume data. Red vertical lines indicates detected breakpoint.

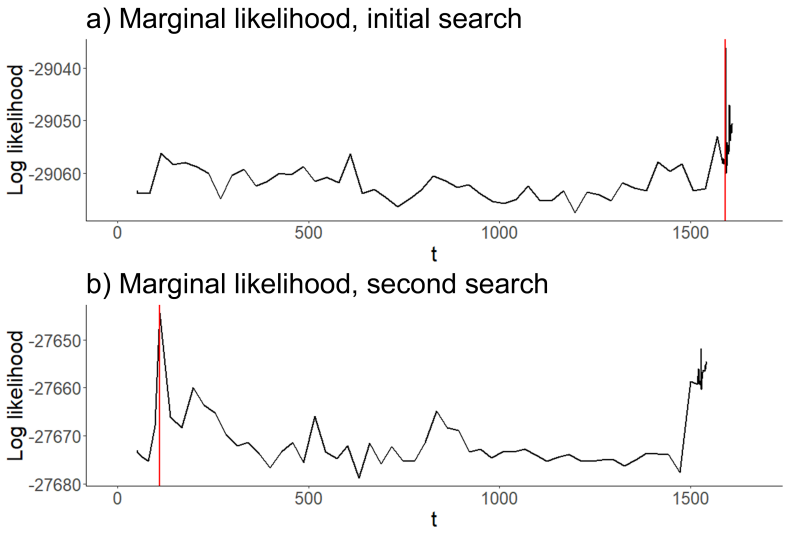


Figure 16: Marginal likelihood given change point when applying breakpoint detection data on the TSLA-stock trading volume data. Found breakspoints are displayed as vertical lines.

the number of days we have data from it is surprising how stable levels the correlation seems to be. The mode of the found correlation coefficients for the respective parts in the final model were $\rho_1 = 0.555, \rho_2 = 0.443$ and $\rho_3 = 0.564$. The first and the third segment of the count observations seem to match by having relatively equal correlation, while the second has lower levels of correlation between days. We also note, indicated from the simulations, that we are in area of correlation differences where we have power that likely is far under 50%, which hurts our ability to detect more change points, and potential other breakpoints might pose as false negatives.

Figure 16 displays the likelihood given a breakpoint at a given position for the different searches. In general these seems relatively flat except a few spikes, this indicates that there are no more breaks as earlier sections discuss. We also display the mode of the estimated latent AR(1) time series in Figure 17, it seems to mirror the count data, with that the latent AR(1) might be a bit overfit. A possibility to improve the model is thus to change the PC prior used for correlation, we however doubt that doing this would have any significant effect on the detected breakpoints.

6 Discussion and conclusion

There are plenty of ways to discover and classify breakpoints in time series of counts. The methods implemented in this text seemed to perform relatively well and achieved high power for the easiest and medium difficult cases. The marginal likelihood and the DIC statistic proved to be the best at detecting change in the correlation structure and comparing posterior marginals for the

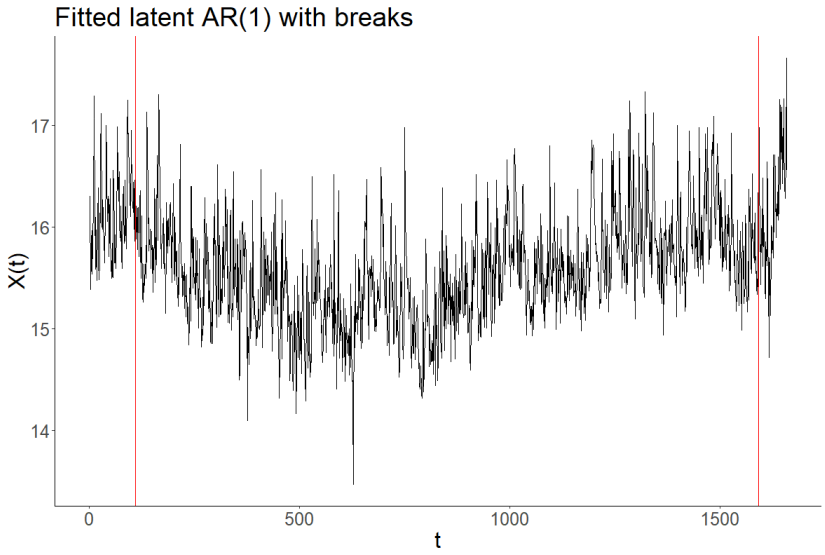


Figure 17: Latent AR(1) in final model with breakpoints of TSLA trade volume model. Found breakspoints are displayed as vertical lines.

variance parameter with the L2 norm seemed to be best at detecting breaks in variances.

The method naturally performed worse when breaks were small. A possible remedy to this could be to assume that parameters in different partitions are from same distribution but of different samples. This might allow detection of finer differences. Another extension of the method would be to test for both changes in variance and structure at the same time.

The method implemented is relatively slow and could be improved by rewriting the INLA scheme to utilize that one does several evaluations with data that only differs by some summation. This could for instance be done by dynamically storing point likelihoods as these are the same, in our implementation these were also revaluated at each break test. Another option might be to cut any calculation of posteriors of the latent variables and only calculate the posterior marginals of the hyperparameters.

On the real data application, the method produced reasonable breaks, but again might have been too strict in allowing for breaks. An improvement on the recursive partitioning could also be done by considering family wise error rate.

One could introduce some way of connecting areas where parameters are similar, i.e. in the TSLA stock case trading might occur in the same pattern after a product release or some similar event that causes trading pattern to stay the same. A possibility is to create a time series that assumes that these periods follow the same model and to use that to improve parameter estimation. A less strict version of this is to say that parameters in these periods come from the same distribution.

Time series is of course not the only model type that breakpoint detection can be applied to. Robert B. Gramacy and Lee (2012) for instance create a framework for detecting breakpoints in Gaussian Processes using MCMC. While

Zeileis et al. (2002), Zeileis (2006) and Zeileis and Hornik (2007) create a more general breakpoint detection method. An idea could be to apply the methods presented here on other types of models.

References

Abujiya, M. R. (2017). New Cumulative Sum Control Chart for Monitoring Poisson Processes. *IEEE Access*, 5:14298–14308.

Andrews, D. W. K. (1993). Tests for Parameter Instability and Structural Change With Unknown Change Point. *Econometrica*, 61(4):821–856. Publisher: [Wiley, Econometric Society].

Brown, R. L., Durbin, J., and Evans, J. M. (1975). Techniques for Testing the Constancy of Regression Relationships over Time. *Journal of the Royal Statistical Society. Series B (Methodological)*, 37(2):149–192.

Chen, C. W. S. and Lee, S. (2016). Generalized Poisson autoregressive models for time series of counts. *Computational Statistics & Data Analysis*, 99:51–67.

Doukhan, P. and Kengne, W. (2013). Inference and testing for structural change in time series of counts model. *arXiv:1305.1751 [math, stat]*. arXiv: 1305.1751.

Hjort, N. L. and Koning, A. (2002). Tests For Constancy Of Model Parameters Over Time. *Journal of Nonparametric Statistics*, 14(1-2):113–132. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10485250211394.

Hubin, A. and Storvik, G. (2016). Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA). *arXiv:1611.01450 [stat]*. arXiv: 1611.01450.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning*, volume 103 of *Springer Texts in Statistics*. Springer New York, New York, NY.

Jung, R., Kukuk, M., and Liesenfeld, R. (2005). Time Series of Count Data: Modelling and Estimation. Working Paper 2005-08, Economics Working Paper.

Kullback, S. (1968). *Information Theory and Statistics*. Dover Publications Inc., Mineola, New York.

Lee, S., Lee, Y., and Chen, C. W. S. (2016). Parameter change test for zero-inflated generalized Poisson autoregressive models. *Statistics*, 50(3):540–557.

Martino, S. and Riebler, A. (2019). Integrated Nested Laplace Approximations (INLA). *arXiv:1907.01248 [stat]*. arXiv: 1907.01248.

Page, E. S. (1954). CONTINUOUS INSPECTION SCHEMES. *Biometrika*, 41(1-2):100–115. Publisher: Oxford Academic.

Robert B. Gramacy and Lee, H. K. (2012). Bayesian Treed Gaussian Process Models With and Application to Computer Modeling. *Journal of the American Statistical Association*, pages 1119–1130.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392. _eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9868.2008.00700.x.

Simpson, D. P., Rue, H., Martins, T. G., Riebler, A., and Sørbye, S. H. (2015). Penalising model component complexity: A principled, practical approach to constructing priors. *arXiv:1403.4630 [stat]*. Reporter: arXiv:1403.4630 [stat] arXiv: 1403.4630.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Linde, A. V. D. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):583–639. _eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/1467-9868.00353.

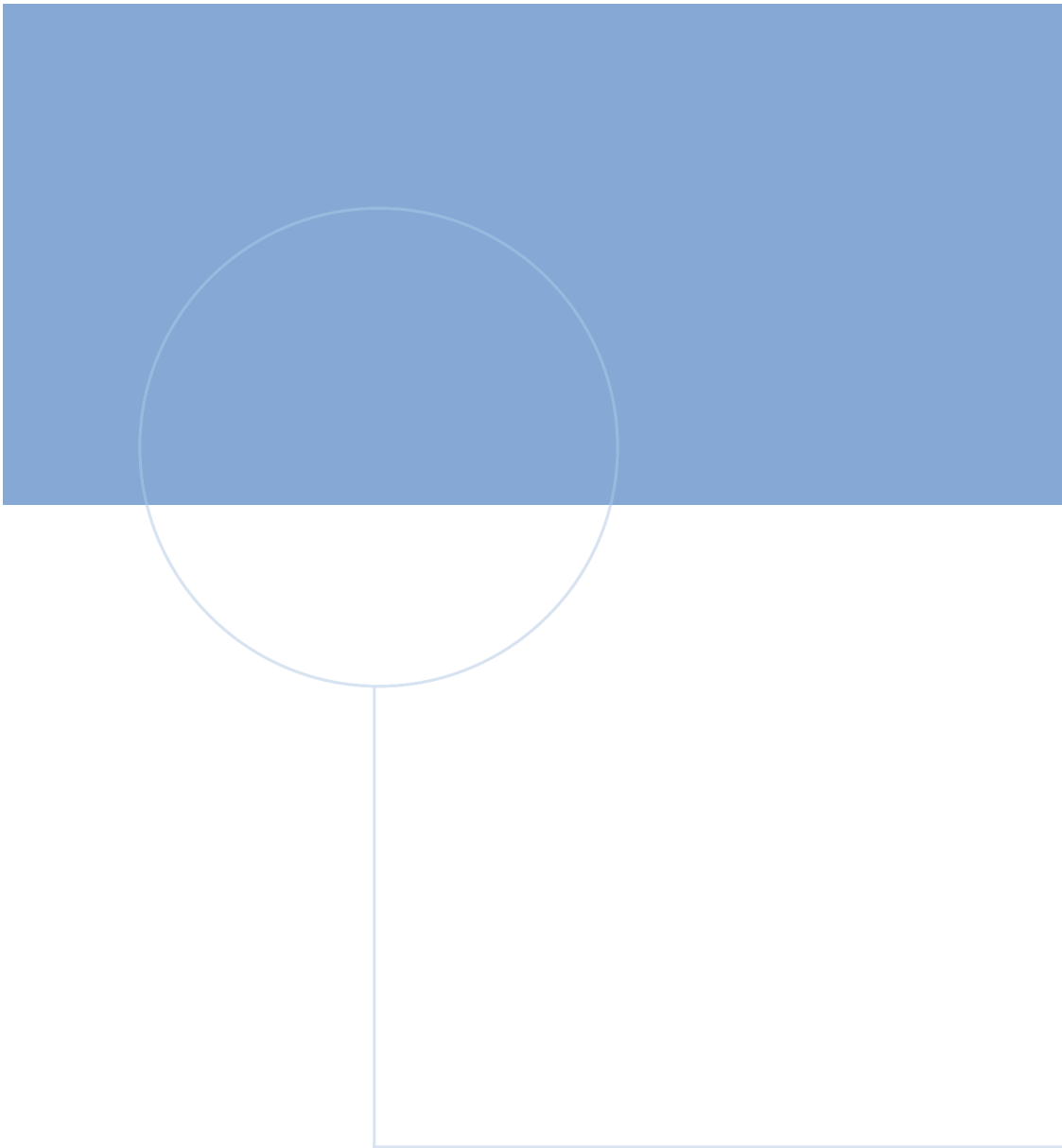
Tsay, R. S. (1988). Outliers, level shifts, and variance changes in time series. *Journal of Forecasting*, 7(1):1–20.

Winkelmann, R. (2008). *Econometric Analysis of Count Data*. Springer-Verlag, Berlin Heidelberg, 5 edition.

Zeileis, A. (2006). Implementing a class of structural change tests: An econometric computing approach. *Computational Statistics & Data Analysis*, 50(11):2987–3008. Number: 11 Reporter: Computational Statistics & Data Analysis.

Zeileis, A. and Hornik, K. (2007). Generalized M-fluctuation tests for parameter instability. *Statistica Neerlandica*, 61(4):488–508. Number: 4 Reporter: Statistica Neerlandica.

Zeileis, A., Leisch, F., Hornik, K., and Kleiber, C. (2002). strucchange: An R Package for Testing for Structural Change in Linear Regression Models. *Journal of Statistical Software*, 7(1):1–38. Number: 1 Reporter: Journal of Statistical Software.



NTNU

Norwegian University of
Science and Technology