Change Points

Nils Hallerfelt

Mars 2023

Introduction

In this report we are going to take a closer look at coal mine disasters in Great Britain between 1658 and 1980. The data is taken from http://www.cmhrc.co.uk/site/disasters/index.html . Since the dataset span over more than 300 years it is natural to assume that the conditions have changed over the years. Opening of new mines and closing of old mines, development of new technology and varying demand for coal are some of the factors that can cause a change in disaster intensity.

Stating the problem and notation

Let $t_1 = 1658$ and $t_{d+1} = 1980$ denote the fixed start and end point of the dataset. We denote the breakpoints by t_i for $i = 2, \dots, d$. Its natural to collect end points and break points in a vector $\mathbf{t} = (t_1, \dots, t_{d+1})$. The disaster intensity in each interval $[t_i, t_{i+1})$ is λ_i, and we let $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_d)$.

The data consists of time continuous data where $\boldsymbol{\tau} = (\tau_1, \dots, \tau_n)$ denotes the time points for the $n = 751$ disasters. We model the data on the interval $t_1 \leq t \leq t_{d+1}$. using an inhomogeneous Poisson process with intensity

$$\lambda(t) = \sum_{i=1}^{d} \lambda_i \mathbf{1}_{[t_i, t_{i+1})}(t)$$

This implies that the times between accidents have an exponential distribution with intensity λ_i for accidents in the interval $[t_i, t_{i+1})$. The time between the last accident in interval i and the breakpoint t_{i+1} is an observation of a truncated exponential random variable. All we know here is that it is larger than the time span left in the interval.

From the time points of the disasters we compute

$$n_i(\boldsymbol{\tau}) = \text{number of disasters in the sub-interval } [t_i, t_{i+1}) = \sum_{j=1}^{n} \mathbf{1}_{[t_i, t_{i+1}]}(\tau(j))$$

We put a $\Gamma(2, \theta)$-prior on the intensities with a $\Gamma(2, \Psi)$-hyperprior on θ, where Ψ is a fixed hyperparameter that needs to be specified. In addition, we put a prior on the breakpoints:

$$f(\mathbf{t}) \propto \begin{cases} \prod_{i=1}^{d} (t_{i+1} - t_i), & \text{for } t_1 < t_2 < \cdots < t_d < t_{d+1} \\ 0, & \text{otherwise} \end{cases} .$$

This implies that

$$f(\boldsymbol{\tau}|\boldsymbol{\lambda},\boldsymbol{\tau}) = \exp\big(-\sum_{i=1}^{d}\lambda_i(t_{i+1}-t_i)\big)\prod_{i=1}^{d}\lambda_i^{n_i(\boldsymbol{\tau})} \tag{1}$$

To sample from the posterior $f(\theta,\boldsymbol{\lambda},\mathbf{t}|\boldsymbol{\tau})$ we will construct a hybrid MCMC algorithm as follows. All components except the breakpoints $\boldsymbol{t}$ can be updated using Gibbs sampling. To update the breakpoints we use a Metropolis-Hastings (MH) sampler. Note that all proposals which change the order of the breakpoints should have zero acceptance probability due to the assumption of order points in the model setup. There are several possible proposal distributions for the MH step. We will look at two slightly different approaches.

1. *Random walk proposals one at a time:* Update one breakpoint at a time. For each breakpoint t_i we generate a candidate $t_i^* = t_i + \epsilon$, with $\epsilon \sim \mathbf{U}(-R,R)$ and $R = \rho(t_{i+1}-t_{i-1})$.

2. *Random walk proposals all at once:* The random walk proposal where we suggest a new position for all the breakpoints at once. Each of the points is now proposed as $t_i^* = t_i+\epsilon$, with $\epsilon \sim \mathbf{U}(-\rho,\rho)$. Now either accept the entire vector and move the breakpoints or reject the move and stay at the same breakpoints.

Marginal Posteriors and their Distributions

In this section we will investigate the conditional probabilities

1. $f(\theta|\boldsymbol{\lambda},\boldsymbol{t},\boldsymbol{\tau})$

2. $f(\boldsymbol{\lambda}|\theta,\boldsymbol{t},\boldsymbol{\tau})$

3. $f(\boldsymbol{t}|\theta,\boldsymbol{\lambda},\boldsymbol{\tau})$

We begin by writing the joint distribution of our random variables as:

$$f(\theta,\boldsymbol{\lambda},\boldsymbol{t},\boldsymbol{\tau}). \tag{2}$$

We can rewrite the joint distribution in 2 using the chain rule for probabilities

$$\begin{aligned} &f(\theta,\boldsymbol{\lambda},\boldsymbol{t},\boldsymbol{\tau}) = && (3)\\ &f(\boldsymbol{\tau}|\theta,\boldsymbol{\lambda},\boldsymbol{t})\cdot && (4)\\ &f(\boldsymbol{t}|\theta,\boldsymbol{\lambda})\cdot && (5)\\ &f(\boldsymbol{\lambda}|\theta)\cdot && (6)\\ &f(\theta) && (7)\end{aligned}$$

and treat each conditional distribution separately for a moment.

Firstly consider 4. As τ does not depend on θ we can use 1:

$$f(\boldsymbol{\tau}|\theta,\boldsymbol{\lambda},\boldsymbol{t}) = f(\boldsymbol{\tau}|\boldsymbol{\lambda},\boldsymbol{t}) = \exp\big(-\sum_{i=1}^{d}\lambda_i(t_{i+1}-t_i)\big)\prod_{i=1}^{d}\lambda_i^{n_i(\boldsymbol{\tau})}. \tag{8}$$

Secondly consider 5. As $\boldsymbol{t}$ does not depend on either $\boldsymbol{\lambda}$ or θ:

$$f(\boldsymbol{t}|\theta,\boldsymbol{\lambda}) = f(\boldsymbol{t}) \propto \begin{cases} \prod_{i=1}^{d}(t_{i+1}-t_i), & \text{for } t_1 < t_2 < \cdots < t_d < t_{d+1} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Thirdly, consider 6. We have a $\Gamma(2,\theta)$ prior on λ, so:

$$f(\boldsymbol{\lambda}|\theta) = \prod_{i=1}^{d} \frac{\theta^2}{\Gamma(2)} \lambda_i \exp\left(-\theta\lambda_i\right) \quad \boldsymbol{\lambda} \geq 0. \tag{10}$$

Lastly we consider 7. We have set a $\Gamma(2,\Psi)$-hyperprior on θ, so:

$$f(\theta) = \frac{\Psi^2}{\Gamma(2)} \theta \exp\left(-\Psi\theta\right) \tag{11}$$

We can now write the joint the joint probability by multiplying equations 8, 9, 10, 11, and an unknown normalizing constant c. Furthermore, we can now determine the conditional probabilities up to a normalizing constant by treating the conditioning variables (independent from the conditional variable) as constants:

1. $f(\theta|\boldsymbol{\lambda},\boldsymbol{t},\boldsymbol{\tau}) \propto \theta^{2d+1} \exp\left(-\theta(\Psi + \sum_{i=1}^{d}\lambda_i\right)$
2. $f(\boldsymbol{\lambda}|\theta,\boldsymbol{t},\boldsymbol{\tau}) \propto \prod_{i=1}^{d} \lambda_i^{n_i(\tau)+1} \exp\left(-\lambda(t_{i+1}-t_i+\theta)\right)$
3. $f(\boldsymbol{t}|\theta,\boldsymbol{\lambda},\boldsymbol{\tau}) \propto \prod_{i=1}^{d} \lambda_i^{n_i(\tau)}(t_{i+1}-t_i) \exp\left(-(t_{i+1}-t_i)\lambda_i\right)$

We can see that $f(\theta|\boldsymbol{\lambda},\boldsymbol{t},\boldsymbol{\tau})$ is a $\Gamma(d+2, \Psi + \sum_{i=1}^{d}\lambda_i)$, and that $f(\boldsymbol{\lambda}|\theta,\boldsymbol{t},\boldsymbol{\tau})$ is a $\prod_{i=1}^{d}\Gamma(n_i(\boldsymbol{\tau})+2, (t_{i+1}-t_i+\theta))$. However $f(\boldsymbol{t}|\theta,\boldsymbol{\lambda},\boldsymbol{\tau})$ does not have a known distribution.

Constructing a Hybrid MCMC-algorithm to Sample From the Posterior $f(\theta,\boldsymbol{\lambda},\boldsymbol{t}|\boldsymbol{\tau})$

The types of Markov chains we will work with will be constructed such that the Markov chain is irreducible and aperiodic, resulting in convergens to the limiting stationary distribution of the chain. Then, for sufficiently long chains, a realization from this chain will have approximate marginal distribution f that is the desired unknown.

In the above section we could see that $f(\theta|\boldsymbol{\lambda},\boldsymbol{t},\boldsymbol{\tau})$ is a $\Gamma(d+2, \Psi + \sum_{i=1}^{d}\lambda_i)$, and that $f(\boldsymbol{\lambda}|\theta,\boldsymbol{t},\boldsymbol{\tau})$ is a $\prod_{i=1}^{d}\Gamma(n_i(\boldsymbol{\tau})+2, (t_{i+1}-t_i+\theta))$. These conditional posteriors are thus easy to sample from using Gibbs sampling.

Gibbs sampling is a probabilistic algorithm that can be used to approximate the joint probability distribution of a set of variables, even if it is complex or high-dimensional. The algorithm is based on the idea of MCMC framework, sampling from the conditional distributions of each variable given the current values of the other variables. For our case, distributions are $f(\theta|\boldsymbol{\lambda},\boldsymbol{t},\boldsymbol{\tau})$ and $f(\boldsymbol{\lambda}|\theta,\boldsymbol{t},\boldsymbol{\tau})$ that are Γ-distributed.

To start the algorithm, an initial value is chosen. We have here used a $\Gamma(2, \Psi)$-hyperprior on θ to initiate.Then, in each iteration, a single variable is chosen at random and its value is updated based on the current values of the other variables. The new value is sampled from the conditional distribution of the variable given the current values of the other variables. The other variables are held constant during this step. This process is repeated for each variable in turn, cycling through the variables multiple times to create a chain of samples.

As the algorithm progresses, the samples generated by the Gibbs sampler converge to the true distribution of the variables. This convergence is guaranteed if the algorithm is run for a sufficient number of iterations and if certain conditions on the conditional distributions are met (e.g., they are well-defined and can be easily sampled from) [Computational Statistics, Givens and Hoeting, page 209-212].

For our purpose there is however complications. To complete one cycle $\theta \xrightarrow{Gibbs} \boldsymbol{\lambda} \xrightarrow{NotGibbs} \boldsymbol{t}$ we can not use Gibbs to generate $\boldsymbol{t}$, as this distribution is hard to sample from even with sampled λ. We thus resort to using the Metropolis-Hasting-algorithm (MH) in doing so.

The Metropolis-Hastings algorithm works by iteratively generating a sequence of samples from a target distribution by randomly proposing a new sample and then accepting or rejecting the proposed sample based on a probability ratio. Specifically, the algorithm proceeds as follows:

1. Choose an initial state or sample for the Markov Chain.

2. Propose a new sample by selecting a candidate value from a proposal distribution:

$$t^* \sim g(\cdot|t_i)$$

3. Calculate the acceptance ratio, which is the ratio of the probability of the proposed sample under the target distribution to the probability of the current sample under the target distribution. This ratio is often called the Metropolis-Hastings ratio:

$$\mathbf{R}(t_i, t^*) = \frac{f(t^*)g(t_i|t^*)}{f(t_i)g(t^*|t_i)}$$

4. Generate a uniform random number between 0 and 1. If the random number is less than the acceptance ratio, then accept the proposed sample as the next state of the Markov Chain. Otherwise, reject the proposed sample and keep the current state.

5. Repeat steps 2-4 for a large number of iterations to generate a samples from the target distribution.

The Metropolis-Hastings algorithm can be used to sample from any distribution, regardless of its shape or complexity, as long as the target distribution is known up to a constant of proportionality. However, the algorithm can be sensitive to the choice of proposal distribution, and other hyperparameters may require tuning to achieve good performance.

In this implementation a g = *random walk proposal one at a time* proposal will be used, explained in the introduction. Thus the proposal is symmetric, so that $\mathbf{R}(t_i, t^*)$ reduces to:

$$\mathbf{R}(t_i, t^*) = \frac{f(t^*)}{f(t_i)}.$$

Furthermore, the random walk MC-chain that gets constructed using g, as the support of f (our target distribution) is connected over the support of g which does not have a support exceeding that of f. This means that the Metropolis-Hasting ratio will not be non-zero, allowing the random walk move over the target distribution, giving probable irreducibility. The chain should also be aperiodic, as it is a random walk. However it could maybe with bad luck be forced by the target distribution into periodicity, but unfortunately we cannot motivate this further.

Results

Using the described algorithm, we can now investigate its behaviour for different amounts of breakpoints. We do this by running it with $\Psi = 1$. All runs are done for 500 000 iterations. We can see that the algorithm finds

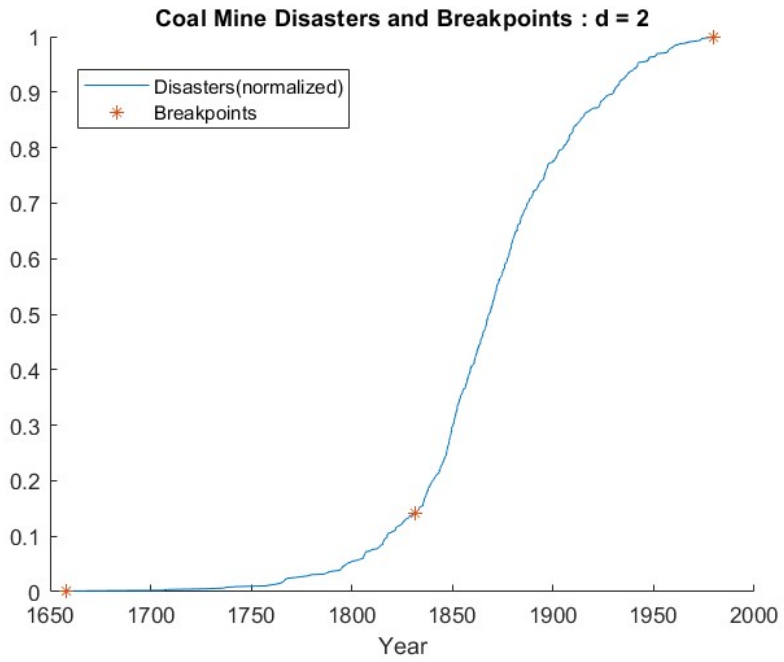

Figure 1: Normalized coal mine disasters plotted with only one breakpoint ($d = 2$) in between the endpoints.

one breakpoint around 1830 with $d = 2$.

For $d = 3$ it finds one breakpoint in the same region as for $d = 2$, while also placing one of the breakpoints in the region 1870.

For $d = 4$ the it places two breakpoints close together in the region of 1830, while also placing one in around 1910. This breakpoint corresponds to the one found around 1870, however this is a rather large fluctuation and could mean that the stationary distribution have not been tightly identified, or that early values in the sampling

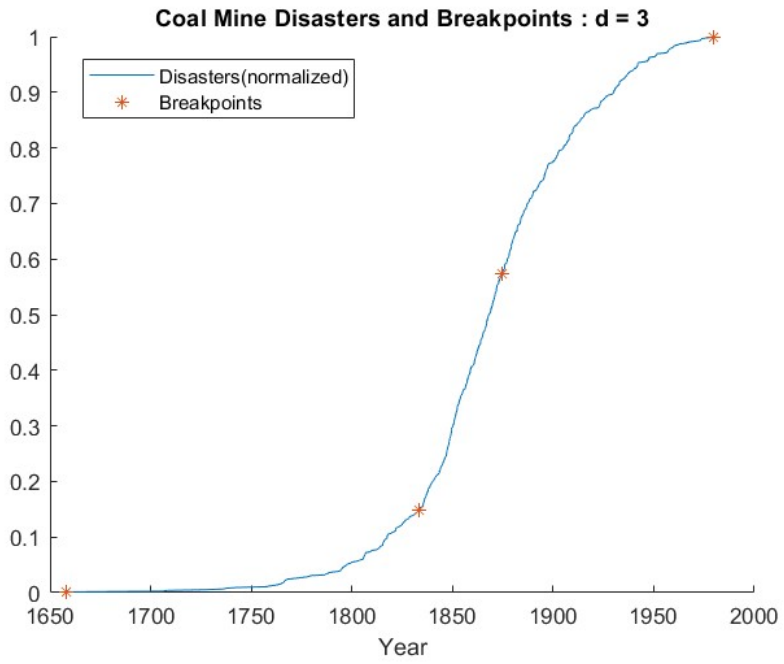

Figure 2: Normalized coal mine disasters plotted with only one breakpoint ($d = 3$) in between the endpoints.

procedure has a large impact on this breakpoint. This indicates that further analysis with burn-in should be done.

For $d = 5$ the algorithm clusters the breakpoints close to the late endpoint, and does not behave in a desirable way. This could be due to that the constructed chain no longer is irreducible.

Hyperprior-paramter Ψ

We can investigate the effect of Ψ on the conditional posteriors. This can be done using the KL-divergence for different values of Ψ. Here we have taken $\Psi = 1$, the value used in the simulations, as the base. Then the absolute KL-divergence for all the posteriors has been calculated, with respect to their base. This has been done for $\Psi = 2, 3, 4, 5, 6$, and the results can be seen in the figures 5, 6, 7.

As we can see, the further away from the basecase Ψ gets the larger the KL-divergence gets for $f(\theta|\boldsymbol{\lambda}, \boldsymbol{t}, \boldsymbol{\tau})$ and $f(\boldsymbol{\lambda}|\theta, \boldsymbol{t}, \boldsymbol{\tau})$, while its more fluctuating for t. This does also indicate that that Ψ have a larger effect on $f(\theta|\boldsymbol{\lambda}, \boldsymbol{t}, \boldsymbol{\tau})$ and $f(\boldsymbol{\lambda}|\theta, \boldsymbol{t}, \boldsymbol{\tau})$ then it does for $f(\boldsymbol{\lambda}|\theta, \boldsymbol{t}, \boldsymbol{\tau})$. In retrospect, these plots should have been normalized for an easier and better comparison.

Sensitivity in Mixing and Posteriors with regards to choice of ρ

To get good mixing we want a acceptance rate of around 30%. Because we use the same ρ for all different number number of breakpoints we are not always able to get good mixing. This is because ρ influences the ac-

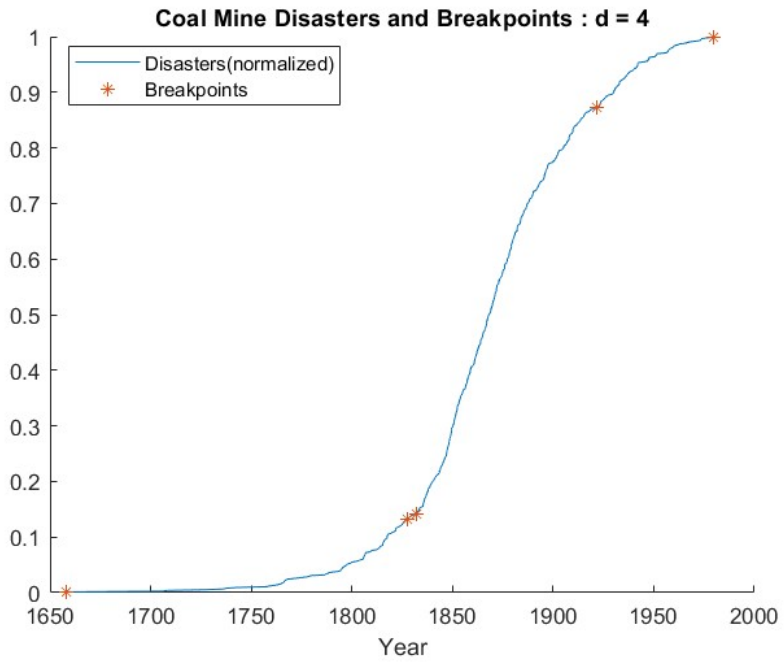

Figure 3: Normalized coal mine disasters plotted with only one breakpoint ($d = 4$) in between the endpoints.

ceptance rate in a direct manner, and the same goes for the number of breakpoints. We can see this by fixing the number of breakpoints $d = 3$, and try for $\rho = (0.01, 0.1, 0.5, 1)$, yielding $([0.2679, 0.5128], [0.0310, 0.1127], [0.0063, 0.0246][0.0035,$ respectively. We now fix the ρ to the more promising 0.01, and simulate for the breakpoints $d = (2, 3, 4, 5)$, yielding $([0.1325], [0.1630, 0.4863], [0.5334, 0.2199, 0.2668], [0.7538, 0.5998, 0.3415, 0.3945])$. We also examine the dependence of θ, λ, t on ρ as ρ goes from 0.001 to 0.1. The result can be seen in figure 8.

Parametric Bootstrap for the 100-year Atlantic wave

The inverse cumulative distribution function (CDF) of the Gumbel distribution is the quantile function, which maps a given probability to the corresponding value of the random variable. To find the inverse CDF of the Gumbel distribution, we start by setting the CDF of the distribution equal to the given probability p and solving for the corresponding value of the random variable.

The CDF of the Gumbel distribution is given by:

$$F(x) = \exp\left(-\exp\left(-(x-\mu)/\beta\right)\right) \tag{12}$$

To find the inverse CDF of the Gumbel distribution, we set $F(x) = p$ and solve for x:

$$\begin{aligned} p &= exp(-exp(-(x-)/\beta)) \\ -\log(-\log(p)) &= (x-\mu)/\beta \\ x = \mu - \beta\log(-\log(p)) &\implies F^{-1}(u,\mu,\beta) = \mu - \beta\log(-\log(u)) \end{aligned}$$

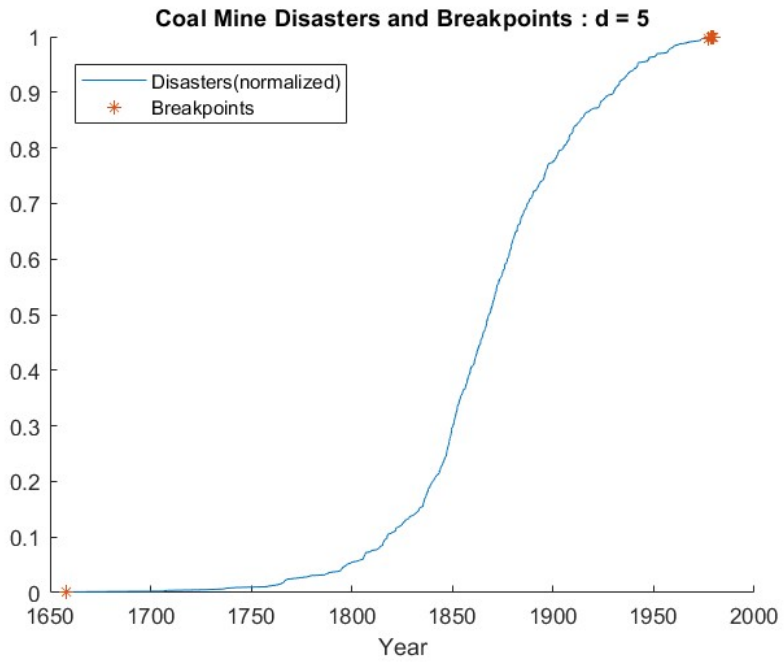

Figure 4: Normalized coal mine disasters plotted with only one breakpoint ($d = 5$) in between the endpoints.

Here we provide 95% double sided confidence bounds on the parameters β and μ using parametric bootstrap. The result can be seen in table 1.

	LB	$\mathbb{E}$	UB
β	1.3990	1.4858	1.5779
μ	4.0210	4.1477	4.2677

Table 1: Estimated 95% confidence bounds and expected value for the parameters β and μ, using parametric bootstrap.

We then estimate an upper one sided 95% confidence bound on the 100-year return value, also using parametric bootstrap.

	$\mathbb{E}$	UB
u	16.5435	17.2240

Table 2: Estimated 95% one sided upper bound and expected value using parametric bootstrap for the 100-year return value

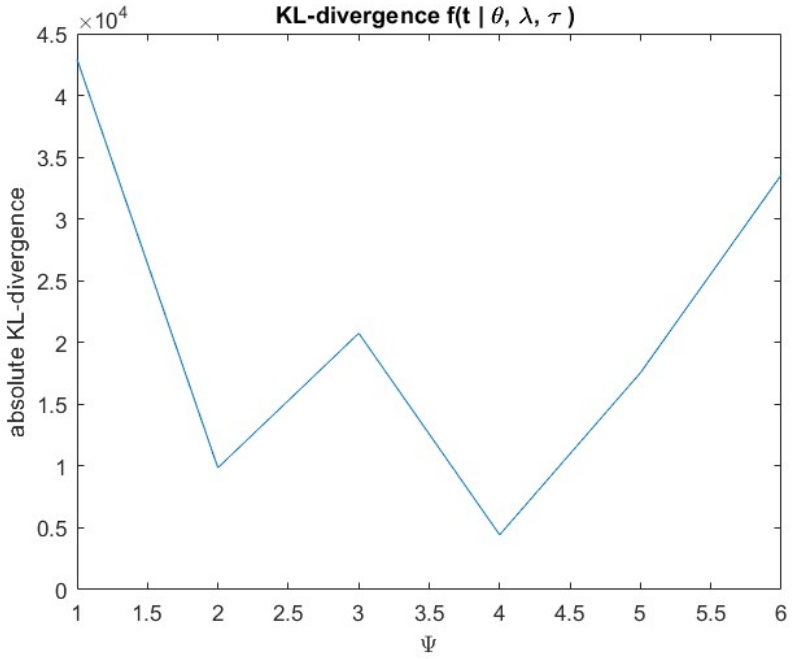

Figure 5: Absolute KL-divergence for $f(\boldsymbol{t}|\theta, \boldsymbol{\lambda}, \boldsymbol{\tau})$ with different hyperprior-parameter Ψ.

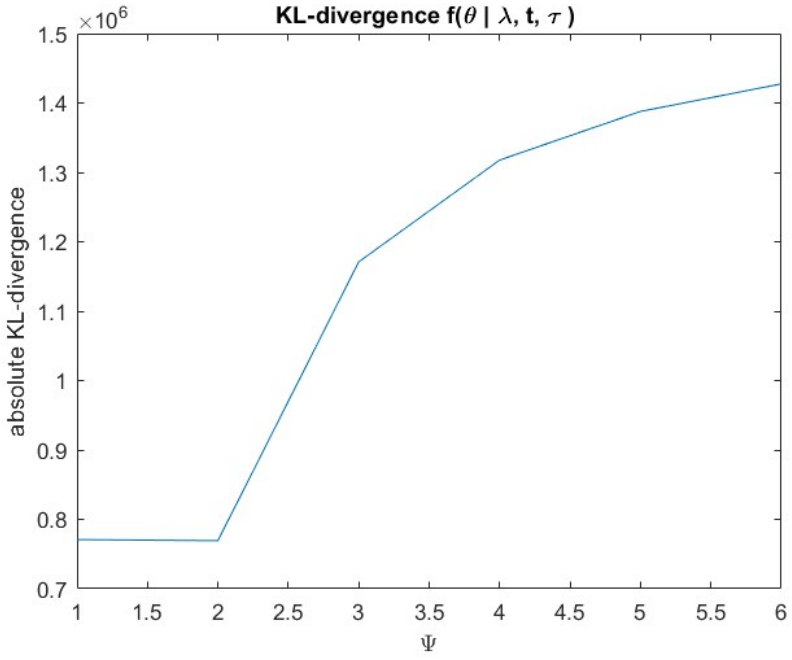

Figure 6: Absolute KL-divergence for $f(\theta|\boldsymbol{\lambda}, \boldsymbol{t}, \boldsymbol{\tau})$ with different hyperprior-parameter Ψ

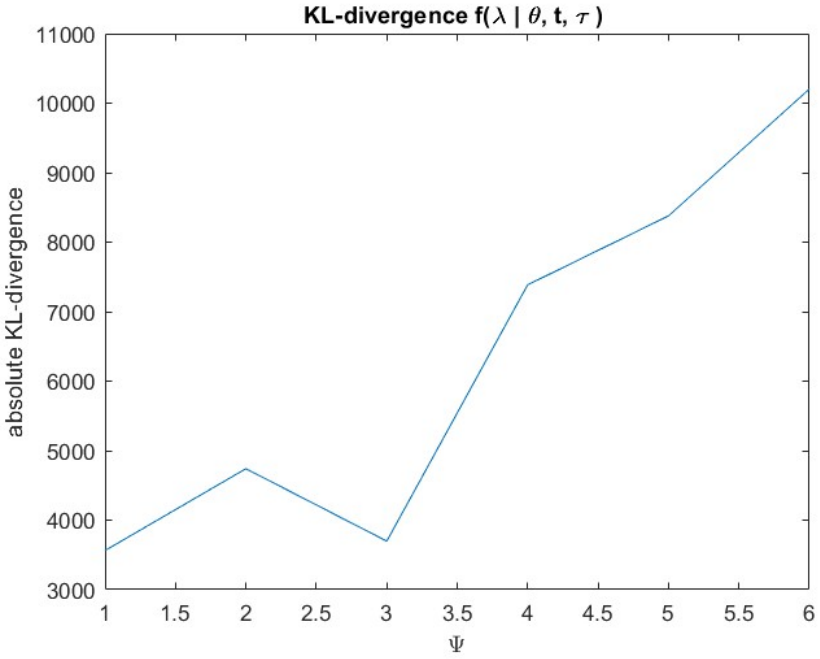

Figure 7: Absolute KL-divergence for $f(\boldsymbol{\lambda}|\theta, \boldsymbol{t}, \boldsymbol{\tau})$ with different hyperprior-parameter Ψ.

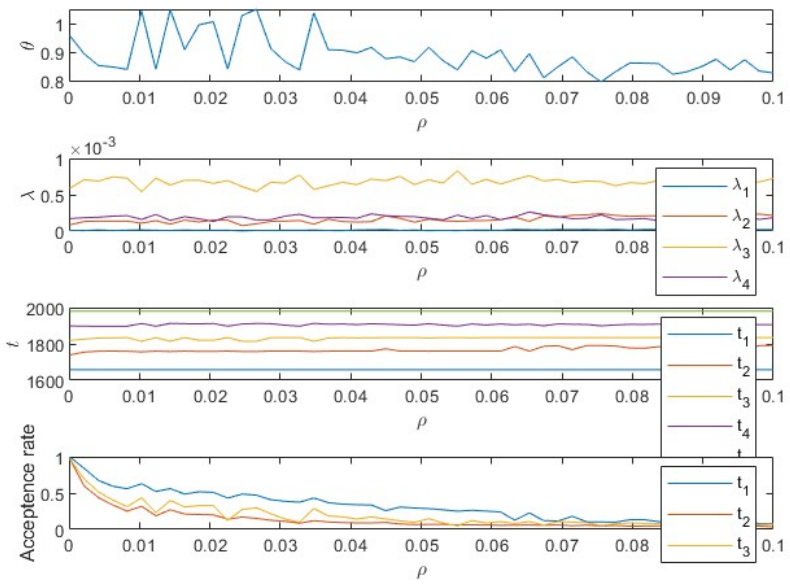

Figure 8: Showing how θ, λ, t and the acceptance rate of the MH changed for different rho from 0.001 to 0.1.