
Bayesian Inference for Change Points in Dynamical Systems with Reusable States—a Chinese Restaurant Process Approach

Florian Stimberg

Artificial Intelligence Group
TU Berlin

Andreas Ruttor

Artificial Intelligence Group
TU Berlin

Manfred Opper

Artificial Intelligence Group
TU Berlin

Abstract

We study a model of a stochastic process with unobserved parameters which suddenly change at random times. The possible parameter values are assumed to be from a finite but unknown set. Using a Chinese restaurant process prior over parameters we develop an efficient MCMC procedure for Bayesian inference. We demonstrate the significance of our approach with an application to systems biology data.

1 Introduction

Continuous time Gaussian Markov processes, e.g. of the Ornstein-Uhlenbeck (OU) type, are broadly used in fields ranging from physics (Weber et al., 2011) to biology (Kostal et al., 2007) to economics and finance (Chan et al., 1992). They often provide a good description of the (short-term) dynamics found in various systems and can be used efficiently for inference purposes. On a larger time scale, however, the parameters of such a model usually do not stay constant due to external or neglected nonlinear effects. Smoothly varying parameters have been successfully modelled using Gaussian processes (Lawrence et al., 2006, Alvarez et al., 2009). But this approach does not take parameters into account, which undergo sometimes rapid changes and stay constant between change points. As such behavior can be observed both in economics (Preis et al., 2011) and systems biology (Oppel et al., 2010), it is important to develop appropriate models.

Most approaches to Bayesian inference for change-point models (e.g. Fearnhead and Liu, 2011, Giordani and Kohn, 2008) only consider direct independent noisy observations of the change-point process. But often in dynamical systems we only get observations from an intermedi-

ary stochastic process with latent parameters modelled by a change-point process. Additionally, parameters for each time interval between neighboring change points, which is called a *segment* (Fearnhead and Liu, 2011), are usually assumed to be drawn independently from a continuous distribution. Thus each parameter value is only used once with probability 1. In contrast, there are systems, where parameters of the dynamics switch between a finite, but unknown number of discrete states (Barber, 2006, Oppel et al., 2010). These, however, occur with unknown probabilities. Algorithms taking this hidden distribution of parameter values into account can discover additional structural information and achieve more efficient estimates by joining segments belonging to the same state. Therefore we propose to model the selection of the parameters for each segment by a Chinese Restaurant process (CRP), because it naturally allows for values to be reused at later points in time. This corresponds to the assumption that the parameter distribution is drawn from a *Dirichlet process*. While CRPs have been used as dynamical models by Fox et al. (2008), their use as part of change-point processes in dynamical systems is new to our knowledge.

In this paper, we present a Markov chain Monte Carlo (MCMC) algorithm which asymptotically samples from the exact posterior density of our hierarchical model. Our sampler is based on the approach used for change-point models without reusable states in Stimberg et al. (2011). The sampler is first tested on synthetic data and then applied to microarray data from yeast cells in metabolic cycles. Afterwards the results are compared to biological knowledge of the system.

2 Generative model

Figure 1 shows the generative model, which we assume for inference on the data sets in this paper. The state of the dynamical system at time t is described by an N -dimensional vector $\mathbf{x}(t)$. Its time evolution is given by the linear stochastic differential equation

$$d\mathbf{x} = (\mathbf{A}(t) - \Lambda\mathbf{x})dt + \Sigma d\mathbf{W} \quad (1)$$

Appearing in Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume 22 of JMLR: W&CP 22. Copyright 2012 by the authors.

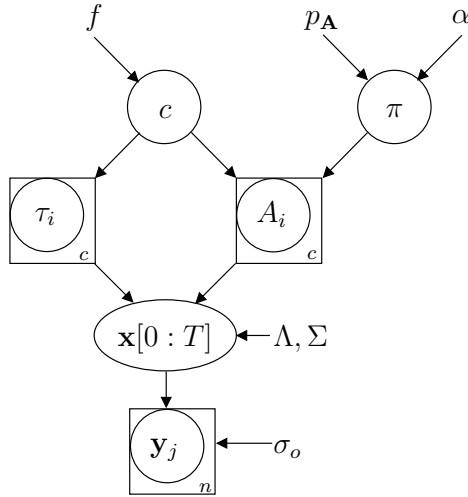


Figure 1: The generative model.

with fixed decay and diffusion parameters

$$\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)^\top, \quad (2)$$

$$\Sigma = \text{diag}(\sigma_1, \dots, \sigma_N)^\top, \quad (3)$$

and a time-dependent function

$$\mathbf{A}(t) = (A_1(t), \dots, A_N(t))^\top, \quad (4)$$

which corresponds to a set of N independent OU processes. We have n observations $\mathbf{Y} = \{y_1, \dots, y_n\}$ of $\mathbf{x}(t)$ at discrete times t_j , $j = 1, \dots, n$, corrupted by i.i.d. Gaussian noise with variance σ_o^2 . Although a path $\mathbf{x}(0 : T)$ is an infinite-dimensional object, it is a Gaussian process and thus can be integrated out exactly as shown in section 3.2. By doing so we obtain directly the likelihood $P(\mathbf{Y}|\mathbf{A}(0 : t), \Sigma, \Lambda, \sigma_o)$ of the observations given the parameters, which is used in the MCMC sampler.

$\mathbf{A}(t)$ is an latent, piecewise constant process. We assume that the number c of its change points in the time interval $[0; T]$ is drawn from a Poisson distribution,

$$c \sim \text{Poisson}(fT), \quad (5)$$

with mean value fT . Conditioned on c their positions $\tau_1, \tau_2, \dots, \tau_c \in [0; T]$ are independently and uniformly distributed random variables:

$$\tau_i \sim U(0, T). \quad (6)$$

After sorting in ascending order these change points divide $[0; T]$ into $c + 1$ segments, where $[\tau_{i-1}; \tau_i]$ denotes the i -th segment, if we additionally define $\tau_0 = 0$ and $\tau_{c+1} = T$. These three steps can be combined to sampling the ordered list $\tau_1, \tau_2, \dots, \tau_c$ as the jump times of a Poisson process with frequency f in the time interval $[0; T]$ (Ross, 1983).

For the duration of each segment the parameter vector $\mathbf{A}(t)$ stays constant, but at each change point a new value is drawn from its distribution π . Since π is unknown, we assume that it comes from a Dirichlet process,

$$\pi \sim \text{DP}(\alpha, p_{\mathbf{A}}), \quad (7)$$

with concentration parameter α and base distribution $p_{\mathbf{A}}$. Integrating out the unknown distribution π leads to a *Chinese Restaurant Process* with the same parameters as the Dirichlet process, which can be sampled sequentially (Teh, 2010). Given the previous segments, the next value

$$\mathbf{A}_{i+1} | \mathbf{A}_1, \dots, \mathbf{A}_i \sim \frac{1}{\alpha + i} \left(\alpha p_{\mathbf{A}} + \sum_{k=1}^i \delta_{\mathbf{A}_k} \right) \quad (8)$$

is either sampled from $p_{\mathbf{A}}$ with probability $\alpha/(\alpha + i)$ or an old parameter is reused. In the latter case one element of the sequence $\mathbf{A}_1, \dots, \mathbf{A}_i$ is selected with equal probability.

3 The MCMC Sampler

Applying an Metropolis-within-Gibbs approach we alternate between sampling the latent change-point process $\mathbf{A}(0 : T)$ given the data \mathbf{Y} and the parameters $\Theta = \{\Sigma, \Lambda, \sigma_o\}$ and sampling Θ given \mathbf{Y} and $\mathbf{A}(0 : T)$.

3.1 Sampling the CRP

Our Metropolis-Hastings (MH) step resembles a random walk on the space of all possible paths of $\mathbf{A}(0 : T)$. To create a proposal path we take the previous path and choose one of four different actions:

Shifting the time of a jump The new time of the jump is drawn from a Gaussian distribution centered around the current time and truncated at the neighboring jumps.

Adding a jump The jump time is drawn uniformly from the interval $[0; T]$. With probability p_* the new segment gets a new value \mathbf{A}_i , otherwise it reuses an existing one.

Removing a jump One of the jumps is chosen at random and removed. With equal probability it is decided if the vanished segment uses the \mathbf{A} value of the segment after or before the removed jump.

Switching a state One of the segments is chosen randomly and the value \mathbf{A}_i associated with it is changed to a completely new set with probability p_* , otherwise it is changed to an existing one.

The shift action is important for determining the correct position of the jumps, therefore we choose it with probability 0.5. The probability to add a jump, remove a jump or switch the state of a segment is 0.125, 0.125 and 0.25,

respectively. If a new value of \mathbf{A} is introduced (either by adding a new jump or changing the value of an existing segment) we draw the new value from the posterior of $P(\mathbf{A}_i|Y, \mathbf{A}_{-i}, \Theta)$ as described in section 3.4.

3.2 Data Likelihood

Since we are interested in the posterior over $\mathbf{A}(0 : T)$ we need to be able to compute the likelihood of the observations \mathbf{Y} conditioned on $\mathbf{A}(0 : T)$ and the parameters Θ up to a normalization constant. The OU process $\mathbf{x}(0 : T)$ is integrated out to improve our estimator according to the Rao-Blackwell theorem (see e.g. Robert and Casella, 2004, pp. 130 ff.). Therefore we need to compute

$$\begin{aligned} & P(\mathbf{Y}|\mathbf{A}(0 : T), \Theta) \\ &= \int P(\mathbf{X}|\Theta)P(\mathbf{Y}|\mathbf{X}, \Theta, \mathbf{A}(0 : T)) d\mathbf{X} \\ &= \int \prod_{i=2}^n P(\mathbf{x}_i|\mathbf{x}_{i-1}, \Theta, \mathbf{A}(0 : T))P(\mathbf{y}_i|\mathbf{x}_i) \\ &\quad \times P(\mathbf{y}_1|\mathbf{x}_1) d\mathbf{X}. \end{aligned} \quad (9)$$

If $\mathbf{A}(t)$ is constant between t_{i-1} and t_i then from the solution of the Ornstein-Uhlenbeck process (Gardiner, 2009, p. 73) we know

$$P(\mathbf{x}_i|\mathbf{x}_{i-1}, \Theta, \mathbf{A}(0 : T)) = \mathcal{N}(\mathbf{x}_i|m_{o_i}, v_{o_i}) \quad (10)$$

with

$$\begin{aligned} m_{o_i} &= \mathbf{x}_{i-1} \exp(-\lambda\Delta t) \\ &\quad + \frac{\mathbf{A}(t_i - 1)}{\lambda} (1 - \exp(-\lambda\Delta t)), \\ v_{o_i} &= \frac{\sigma_o^2}{2\lambda} (1 - \exp(-2\lambda\Delta t)), \end{aligned} \quad (11)$$

where $\Delta t = t_i - t_{i-1}$ and $\mathcal{N}(\cdot|m, v)$ is the Gaussian density with mean m and variance v . If $\mathbf{A}(t)$ is not constant between t_{i-1} and t_i , the mean has to be computed iteratively at the jump times. We now can write the factors in (9) as

$$P(\mathbf{y}_i|\mathbf{x}_i) = \mathcal{N}(\mathbf{y}_i|\mathbf{x}_i, \sigma_o^2) \quad (12)$$

and

$$P(\mathbf{x}_i|\mathbf{x}_{i-1}, \Theta, \mathbf{A}(0 : T)) = \mathcal{N}(\mathbf{x}_i|\alpha_i\mathbf{x}_{i-1} + \beta_i, \xi_i) \quad (13)$$

with

$$\alpha_i = \exp(-\lambda\Delta t), \quad (14)$$

$$\beta_i = \frac{\mathbf{A}(t_{i-1})}{\lambda} (1 - \alpha_i), \quad (15)$$

$$\xi_i = \frac{\sigma_o^2}{2\lambda} (1 - \alpha_i^2), \quad (16)$$

if there are no jumps between t_i and t_{i-1} . Otherwise if there are l jumps at $t_{i,1}, \dots, t_{i,l}$ and we define $t_{i-1} = t_{i,0}$

and $t_i = t_{i,l+1}$ then β_i and ξ_i have to be computed iteratively by

$$\beta_{i,j} = \beta_{i,j-1}\alpha_{i,j} + \frac{\mathbf{A}(t_{i,j-1})}{\lambda} (1 - \alpha_{i,j}) \quad (17)$$

$$\xi_{i,j} = \xi_{i,j-1}\alpha_{i,j}^2 + \frac{\sigma_o^2}{2\lambda} (1 - \alpha_{i,j}^2), \quad (18)$$

where $\alpha_{i,j} = \exp(-\lambda(t_{i,j} - t_{i,j-1}))$, $\beta_{i,0} = \xi_{i,0} = 0$, $\beta_{i,k} = \beta_i$, $\xi_{i,k} = \xi_i$, $t_{i,0} = t_{i-1}$ and $t_{i,k} = t_i$. This enables us to solve the integral in (13) by setting

$$z_n = P(\mathbf{Y}|\mathbf{A}(0 : T), \Theta) \quad (19)$$

and computing it recursively through

$$z_i = z_{i-1} \mathcal{N}(\mathbf{y}_i|m_{z_i} - \mathbf{y}_i, v_{z_i} + \sigma_o^2), \quad (20)$$

$$m_{z_i} = m_{i-1}\alpha_i + \beta_i, \quad (21)$$

$$v_{z_i} = \xi_i + v_{i-1}\alpha_i^2, \quad (22)$$

$$m_i = \frac{\sigma_o^2 m_{z_i} + \mathbf{y}_i v_{z_i}}{\sigma_o^2 + v_{z_i}}, \quad (23)$$

$$v_i = \frac{\sigma_o^2 v_{z_i}}{\sigma_o^2 + v_{z_i}}, \quad (24)$$

with $m_1 = \mathbf{y}_1$, $v_1 = \sigma_o^2$ and $z_1 = 1$ as start values.

3.3 Acceptance ratios

The new proposed path $\mathbf{A}^*(0 : T)$ is accepted with the usual MH acceptance probability. Which in this case is

$$P_{\text{Acc}} = \min(1, \Psi_L \Psi_{\text{prior}} \Psi), \quad (25)$$

where the likelihood ratio

$$\Psi_L = \frac{P(\mathbf{Y}|\mathbf{A}^*(0 : T), \Theta)}{P(\mathbf{Y}|\mathbf{A}(0 : T), \Theta)} \quad (26)$$

is computed as described in section 3.2. The prior ratio

$$\Psi_{\text{prior}} = \frac{P(\mathbf{A}^*(0 : T))}{P(\mathbf{A}(0 : T))} \quad (27)$$

and proposal ratio

$$\Psi_Q = \frac{Q(\mathbf{A}(0 : T)|\mathbf{A}^*(0 : T))}{Q(\mathbf{A}^*(0 : T)|\mathbf{A}(0 : T))} \quad (28)$$

follow from (8) and depend on the chosen random walk action:

Shifting the time of a jump Conditioned on the number of jumps the jump times are uniformly distributed. Therefore the prior does not change when we shift the time of a jump. Because the proposal density is truncated at the neighboring change points, it is not symmetrical. If we move a change point from t to t^* and the proposal density is truncated by t_{\min} and t_{\max} , then the ratio is

$$\Psi_{\text{prior}} \Psi_Q = \frac{\Phi(\frac{t_{\max}-t}{\sigma_t}) - \Phi(\frac{t_{\min}-t}{\sigma_t})}{\Phi(\frac{t_{\max}-t^*}{\sigma_t}) - \Phi(\frac{t_{\min}-t^*}{\sigma_t})}, \quad (29)$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

Adding a jump When adding a jump point we have to distinguish between adding a new set of parameters or an already existing one. If we denote the number of segments which use the parameter set \mathbf{A}_i in the old sample by $\#_i$ then for the first case ($\#_i = 0$) the acceptance ratio is

$$\Psi_{\text{prior}}\Psi_Q = \frac{f\alpha p_{\mathbf{A}}(\mathbf{A}_i)}{\alpha + c + 1} \frac{p_{\text{rem}}T}{Q(\mathbf{A}_i)(c+1)p_{\text{add}}p_*}, \quad (30)$$

where k is the number of parameter sets and c the number of jumps *before* applying the proposal action. p_{add} and p_{rem} are the probabilities to choose the action to add and remove a jump, respectively and $Q(\mathbf{A}_i)$ is the density from which the new \mathbf{A}_i is drawn (in our case the posterior $P(\mathbf{A}_i|Y, \mathbf{A}_{-i}, \Theta)$, see section 3.4). If, on the other hand, the parameter set for the new segment already exists ($\#_i > 0$) the ratio is

$$\Psi_{\text{prior}}\Psi_Q = \frac{\alpha\#_i}{\alpha + c + 1} \frac{p_{\text{rem}}Tk}{(1-p_*)(c+1)p_{\text{add}}}. \quad (31)$$

Removing a jump Removing a change point whose state was the last instance of its kind ($\#_i = 1$) leads to

$$\Psi_{\text{prior}}\Psi_Q = \frac{\alpha + c}{f\alpha p_{\mathbf{A}}(\mathbf{A}_k)} \frac{Q(\mathbf{A}_i)cp_{\text{add}}p_*}{p_{\text{rem}}T}. \quad (32)$$

When \mathbf{A}_i still occurs after removing the change point ($\#_i > 1$) the ratio becomes

$$\Psi_{\text{prior}}\Psi_Q = \frac{\alpha + c}{f(\#_i - 1)} \frac{(1-p_*)cp_{\text{add}}}{kTp_{\text{rem}}}. \quad (33)$$

Switching a state If we switch the current parameter set \mathbf{A}_i of an segment to \mathbf{A}_j there are four different cases to consider:

1. The old parameter set is still used ($\#_i > 1$) and the new parameter set is already active in another segment ($\#_j > 0$)

$$\Psi_{\text{prior}}\Psi_Q = \frac{\#_j}{\#_i}. \quad (34)$$

2. The old parameter set is still used ($\#_i > 1$) and the new parameter set has not been in use ($\#_j = 0$)

$$\Psi_{\text{prior}}\Psi_Q = \frac{\alpha p_{\mathbf{A}}(\mathbf{A}_j)}{\#_i} \frac{(1-p_*)}{Q(\mathbf{A}_j)kp_*}. \quad (35)$$

3. The old parameter set vanishes ($\#_i = 1$) while the new parameter is already used ($\#_j > 0$)

$$\Psi_{\text{prior}}\Psi_Q = \frac{\#_j}{\alpha p_{\mathbf{A}}(\mathbf{A}_i)} \frac{Q(\mathbf{A}_j)kp_*}{(1-p_*)}. \quad (36)$$

4. The old parameter set vanishes ($\#_i = 1$) and the new parameter set has not been in use ($\#_j = 0$)

$$\Psi_{\text{prior}}\Psi_Q = \frac{p_{\mathbf{A}}(\mathbf{A}_j)Q(\mathbf{A}_i)}{p_{\mathbf{A}}(\mathbf{A}_i)Q(\mathbf{A}_j)}. \quad (37)$$

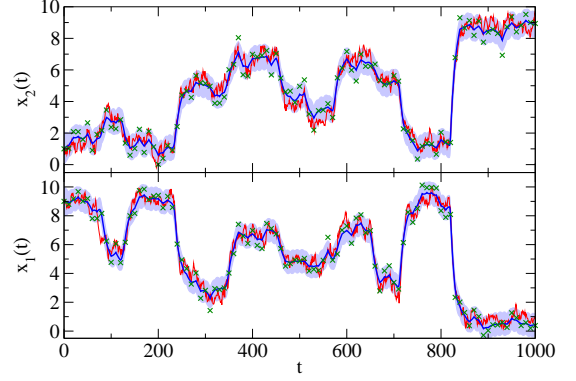


Figure 2: Posterior $\mathbf{x}(0 : T)$ results for a toy data set. The green crosses are the noisy observations and the true process is drawn as a red line. The posterior mean is the blue line surrounded by an confidence interval of two times the standard deviation. The sampler used $\alpha = 1.5$.

3.4 Parameter Sampling

Sampling the parameters is done by another Gibbs sampler which iteratively draws new values for the parameters sets \mathbf{A}_i from the posterior. When looking at the computation of the likelihood in section 3.2 we see that $\mathbf{A}(t)$ is linear in the β_i which is linear in m_{z_i} . This means the different values \mathbf{A}_i are linear in the means of Gaussians which are multiplied to give the likelihood. Therefore we can use the same approach we used for computing the likelihood to propagate the mean and variance depending on \mathbf{A}_i forward. If the base distribution $p_{\mathbf{A}}$ is Gaussian the posterior will be as well and we can easily sample from it. For the applications in this paper we assume that Σ and Λ are known, but if needed they can be sampled by Metropolis-Hastings steps with e.g. a log-Gaussian random walk proposal.

3.5 Samples from the OU Posterior

Since we integrate out $\mathbf{x}(0 : T)$ our sampler does not generate posterior samples for it. However, given the parameters and the CRP the marginal posterior over $\mathbf{x}(0 : T)$ is Gaussian and can be computed exactly without discretization error (see e.g. Archambeau et al., 2007). Depending on the time resolution we are interested in, the computational costs can be quite demanding, therefore the best approach is to compute the posterior only for a thinned out, independent set of samples from the CRP and the parameters.

4 Results

4.1 Synthetic Data

To test our sampler we use a synthetic dataset with a two-dimensional Ornstein-Uhlenbeck process where the hid-

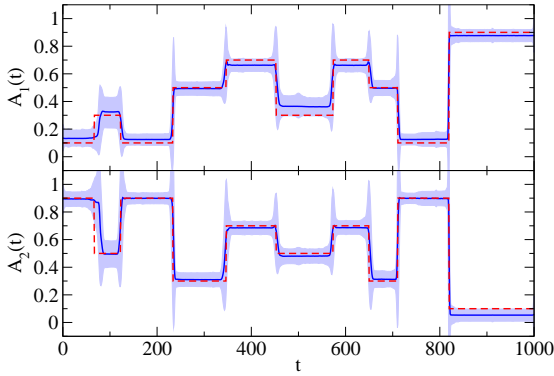


Figure 3: Posterior $\mathbf{A}(t)$ results for a toy data set with α set to 1.5. The red dashed line is the true process and the blue line the posterior mean surrounded by a confidence interval of two times the standard deviation.

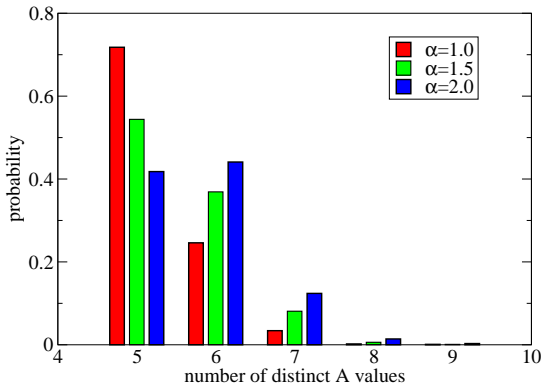


Figure 4: Posterior distribution over the number of different values of \mathbf{A} for the toy data set. The true process has 5 distinct values.

den process $\mathbf{A}(t)$ jumps 9 times and has 5 distinct states: $(0.1, 0.9)$, $(0.3, 0.5)$, $(0.5, 0.3)$, $(0.7, 0.7)$ and $(0.9, 0.1)$. For the base distribution $p_{\mathbf{A}}$ we used a Gaussian with mean and standard deviation 0.5. In figure 2 the data and the posterior distribution for the OU process $\mathbf{x}(t)$ is shown. It can be seen that the posterior fits the true data and successfully ignores outliers. The sampler estimates the latent process $\mathbf{A}(t)$ well, which is clearly visible in figure 3.

Most interesting for the CRP model however is if the sampler determines the number of distinct states in $\mathbf{A}(t)$. When there are 9 jumps the *prior* would have 2.9 states for $\alpha = 1$, 3.5 states for $\alpha = 1.5$ and 4 states for $\alpha = 2$ on average. For these values of α figure 4 shows the *posterior* distribution over the number of states. For $\alpha = 1.0$ and $\alpha = 1.5$ the right number of states (5) is clearly the most probable, while for $\alpha = 2.0$ 6 states is slightly more probable. We can visualize the probability that $\mathbf{A}(t)$ is in the same state at two points in time by a heatmap, as in figure 5. The

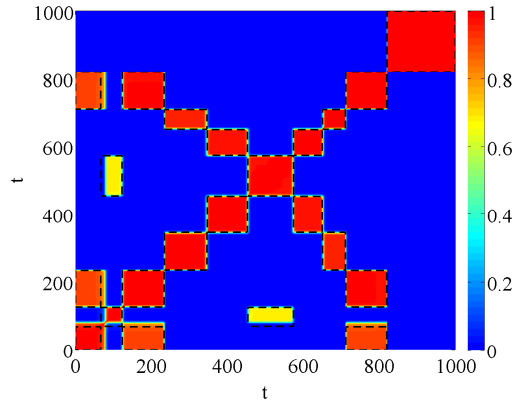


Figure 5: Heatmap results for the toy dataset with $\alpha = 1.5$. The color of a point (t_1, t_2) represents the probability that $\mathbf{A}(t_1) = \mathbf{A}(t_2)$. The black dashed lines surround areas where this is true in the real process.

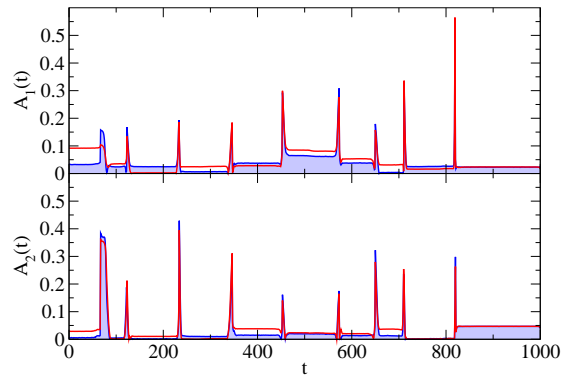


Figure 6: Difference over time between the true $\mathbf{A}(t)$ and the posterior mean of the CRP sampler (blue area, $\alpha = 1.5$) and the change-point sampler (red line) for the toy dataset.

result accurately fits the true heatmap and we can see that e.g. the segment between approximately 70 and 120 and the one between 450 and 570 are only assigned the same state with probability 65%, probably because there are very few observations for the first segment.

4.2 Comparison with a Simplified Change-Point Model

Especially for short segments like this, our CRP model is better suited than a simple change-point model without reusable states. Joining separated instances of a state should improve the parameter inference because there is more data per state. To verify this we applied a similar sampler for a change-point model where the unknown discrete distribution π is replaced by the fixed base distribution $p_{\mathbf{A}}$. Figure 6 shows the absolute difference between

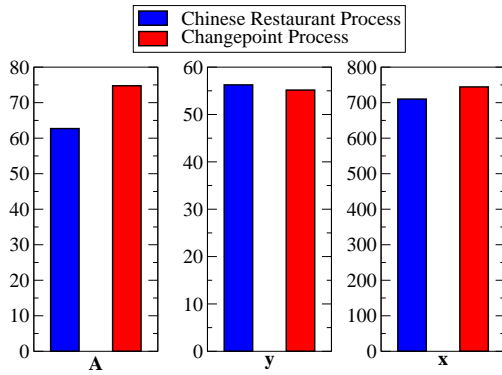


Figure 7: Difference between the true values and simulation results of the CRP sampler with $\alpha = 1.5$ and the change-point sampler on the toy data set.

the true process $\mathbf{A}(t)$ and the posterior mean for both the CRP and the change-point model. The CRP has a lower error most of the time and is—as expected—especially better for smaller segments. If we integrate the error over time we see that the CRP model has an error which is more than 15% smaller than the change-point model’s (see figure 7). Not surprisingly the simplified change-point model has a slightly better fit to the data because it is able to choose the best fitting value of \mathbf{A} in each segment. In contrast to this, the difference between the posterior mean over $\mathbf{x}(t)$ and the true process is around 5% better for the CRP indicating that it more successfully compensates for the observation noise.

Another advantage between the two models is the computation time. While the CRP sampler took roughly 14 hours for half a million samples, the change-point sampler needed over 30 hours for the same number of samples. This is mainly because the change-point model has more distinct states which need to be updated in each iteration. Both samplers are part of the same Matlab program to make the times comparable.

4.3 Transcriptional Regulation in Yeast

As a real dataset we use microarray measurements from yeast cells going through metabolic cycles. The measurements come from Tu et al. (2005) and consist of highly noisy gene expression measurements over time. The yeast cells were forced into metabolic cycles by alternating between starving them and offering glucose.

The expression levels of the genes are regulated by certain proteins, called transcription factors (TF). Normally TFs occur in small copy numbers making them hard to measure. When active they either up- or downregulate the genes’ transcription into mRNA thereby influencing the expression level of the gene. Barenco et al. (2006) modelled

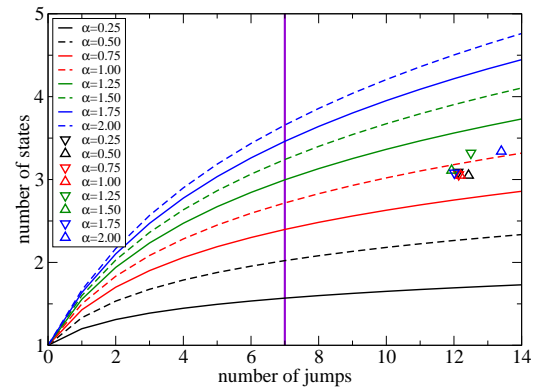


Figure 8: Robustness of the results to different values of the parameter α . The lines depict the mean of the CRP prior while the triangles represent the mean posterior values of the number of jumps and states from simulations with different values of α . The vertical violet line is the mean number of jumps coming from the exponential prior with rate $f = 0.01$.

the expression profile of a gene by an ordinary differential equation (ODE) with the transcriptional regulation represented by changes in the drift. This ODE is the special case of an Ornstein-Uhlenbeck process without diffusion, $\sigma_i^2 = 0$. Applied to our model λ_i represents the degradation rate of the mRNA while $\mathbf{A}_i(t)$ is the transcription rate. Changes in $\mathbf{A}_i(t)$ model switches between different states of TF activity.

We apply our model to a subset of the data consisting of 10 gene expression time series with 36 measurements each. Since the data is averaged over multiple cells the system noise variance σ_i^2 was set to 0 and we use the same degradation rates λ_i and observation noise variance σ_o^2 as Opper and Sanguinetti (2010). The base distribution $p_{\mathbf{A}}$ is a Gaussian distribution with zero mean and 0.25 as standard deviation.

Taking about 7.5 hours on a standard office computer we generated 1 million samples with α set to 1.00. The results predict that 3 distinct states are most likely ($\approx 96\%$) while the rest of the samples indicate at 4 or more states. As can be seen in figure 8 this result is robust to changes in α . Assuming binary states for each TF leads to the conclusion that two transcription factors with states (off,off), (on,off), (off,on) and (on, on) are involved in the transcriptional regulation of these genes.

This fits the biological evidence from Harbison et al. (2004) and Lee et al. (2002) which was combined by Opper and Sanguinetti (2010) to determine that 3 of the proteins are only regulated by the transcription factor *FHL1*, 2 only by *RAP1* and the remaining 5 are regulated by both.

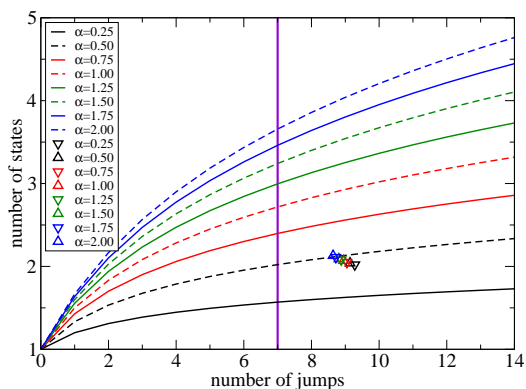


Figure 9: Results on a subset of the yeast data with only 3 proteins regulated solely by FHL1. For a description see figure 8.

We further test our method by doing inference for a dataset of only the 3 proteins which are regulated solely by FHL1. The parameters are the same as for the full dataset but now only two distinct states are found most of the time, corresponding to the active and inactive state of FHL1. Figure 9 again shows this to be robust to changes in α . The periodicity of the transcriptional activity can be seen in figure 10 corresponding to the biological setup of the experiment.

5 Discussion & Outlook

In this paper we consider a model of a continuous time stochastic process with unobserved parameters which undergo sudden changes at random times. The novel aspect of our model is the assumption that the possible parameter values are assumed to be from a finite (typically small) but unknown set. Hence distinct values can occur repeatedly. In a Bayesian approach such a situation can be modelled elegantly by assuming a Chinese restaurant process prior over parameters. We demonstrate an efficient MCMC procedure which for toy models leads to faster and more precise inference results compared to a simple change-point model which draws completely new parameter values for each segment. Applied to a microarray data set, we show that our inference method can be used to estimate the number of transcription factors controlling a set of genes.

While the computational times for the datasets presented here seem rather large it has to be noted that the sampler was implemented in Matlab. We implemented the sampler in Matlab and C++ for a model with a fixed number of states and for the full yeast dataset the C++ program was around 12 times faster than the Matlab implementation. Therefore we expect a similar improvement for the CRP sampler which means 1 million samples for the full yeast data would take roughly 40 minutes.

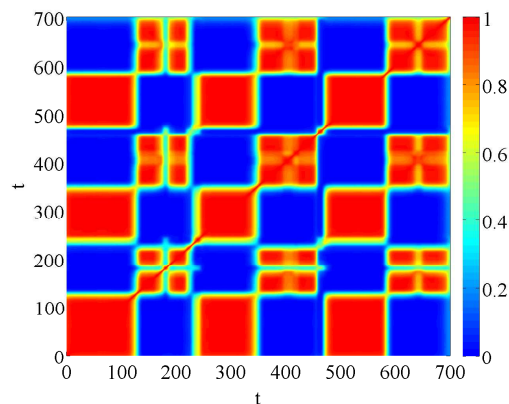


Figure 10: Heatmap for the yeast data with $\alpha = 2$. The color of a point (t_1, t_2) represents the probability that $\mathbf{A}(t_1) = \mathbf{A}(t_2)$.

While our model contains an Ornstein-Uhlenbeck process at the moment, that can be replaced easily by other processes whose likelihood can be computed efficiently, e.g. a Cox-Ingersoll process or a Poisson process where the rate undergoes sudden changes. With these changes the model could be applied to a broader range of data sets, e.g. financial or neurobiological data. One additional extension would be to let $\mathbf{A}(t)$ change only in certain dimensions while others stay the same. This would better represent data where components have different but overlapping change points.

References

- Alvarez, M., Luengo, D., and Lawrence, N. D. (2009). Latent force models. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS)*.
- Archambeau, C., Cornford, D., Opper, M., and Shawe-Taylor, J. (2007). Gaussian process approximations of stochastic differential equations. *Journal of Machine Learning Research Workshop and Conference Proceedings*, 1(1):1–16.
- Barber, D. (2006). Expectation correction for smoothing in switching linear Gaussian state space models. *Journal of Machine Learning Research*, 7:2515–2540.
- Barenco, M., Tomescu, D., Brewer, D., Callard, R., Stark, J., and Hubank, M. (2006). Ranked prediction of p53 targets using hidden variable dynamical modelling. *Genome Biology*, 7(3).
- Chan, K. C., Karolyi, A. G., Longstaff, F. A., and Sanders, A. B. (1992). An empirical comparison of alternative

- models of the short-term interest rate. *Journal of Finance*, 47(3):1209–27.
- Fearnhead, P. and Liu, Z. (2011). Efficient bayesian analysis of multiple changepoint models with dependence across segments. *Statistics and Computing*, 21(2):217–229.
- Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. (2008). An HDP-HMM for systems with state persistence. In *Proc. International Conference on Machine Learning*.
- Gardiner, C. (2009). *Stochastic Methods*. Springer-Verlag Berlin Heidelberg, Berlin, Germany, 4th edition.
- Giordani, P. and Kohn, R. (2008). Efficient bayesian inference for multiple change-point and mixture innovation models. *Journal of Business and Economic Statistics*, 26(1):66–77.
- Harbison, C. T., Gordon, D. B., Lee, T. I. I., Rinaldi, N. J., Macisaac, K. D., Danford, T. W., Hannett, N. M., Tagne, J.-B. B., Reynolds, D. B., Yoo, J., Jennings, E. G., Zeitlinger, J., Pokholok, D. K., Kellis, M., Rolfe, P. A., Takusagawa, K. T., Lander, E. S., Gifford, D. K., Fraenkel, E., and Young, R. A. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104.
- Kostal, L., Lansky, P., and Zucca, C. (2007). Randomness and variability of the neuronal activity described by the ornstein-uhlenbeck model. *Network: Computation in Neural Systems*, 18(1):63–75.
- Lawrence, N. D., Sanguinetti, G., and Rattray, M. (2006). Modelling transcriptional regulation using Gaussian processes. In *Advances in Neural Information Processing Systems 19*.
- Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., Zeitlinger, J., Jennings, E. G., Murray, H. L., Gordon, D. B., Ren, B., Wyrick, J. J., Tagne, J.-B., Volkert, T. L., Fraenkel, E., Gifford, D. K., and Young, R. A. (2002). Transcriptional regulatory networks in *saccharomyces cerevisiae*. *Science*, 298(5594):799–804.
- Opper, M., Rutter, A., and Sanguinetti, G. (2010). Approximate inference in continuous time gaussian-jump processes. In Lafferty, J., Williams, C. K. I., Zemel, R., Shawe-Taylor, J., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 1822–1830.
- Opper, M. and Sanguinetti, G. (2010). Learning combinatorial transcriptional dynamics from gene expression data. *Bioinformatics*, 26(13):1623–1629.
- Preis, T., Schneider, J., and Stanley, H. E. (2011). Switching processes in financial markets. *Proceedings of the National Academy of Sciences USA*, 108(19):7674–7678.
- Robert, C. and Casella, G. (2004). *Monte Carlo statistical methods*. Springer texts in statistics. Springer.
- Ross, S. M. (1983). *Stochastic Processes*. John Wiley & Sons.
- Stimberg, F., Opper, M., Sanguinetti, G., and Rutter, A. (2011). Inference in continuous-time change-point models. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 24*, pages 2717–2725.
- Teh, Y. W. (2010). Dirichlet processes. In *Encyclopedia of Machine Learning*. Springer.
- Tu, B. P., Kudlicki, A., Rowicka, M., and McKnight, S. L. (2005). Logic of the yeast metabolic cycle: temporal compartmentalization of cellular processes. *Science (New York, N.Y.)*, 310(5751):1152–8.
- Weber, C., Radtke, P. K., Schimansky-Geier, L., and Hänggi, P. (2011). Active motion assisted by correlated stochastic torques. *Phys. Rev. E*, 84(1):011132.