An adaptive Markov chain Monte Carlo approach to time series clustering of processes with regime transition behavior

Jana de Wiljes[1], Andrew Majda[2] and Illia Horenko[3]

Abstract

A numerical framework for clustering of time series via a Markov chain Monte Carlo (**MCMC**) method is presented. It combines concepts from recently introduced variational time series analysis and regularized clustering functional minimization (*I. Horenko, SIAM SISC vol. 32(1):62-83*) with **MCMC**. A conceptual advantage of the presented combined framework is that it allows to address the two main problems of the existent clustering methods, e.g., the non-convexity and the ill-posedness of the respective functionals, in a unified way. Clustering of the time series and minimization of the regularized clustering functional is based on generation of samples from an appropriately chosen Boltzmann distribution in the space of cluster affiliation paths using simulated annealing and the Metropolis algorithm. The presented method is applied to sets of generic ill-posed clustering problems and the results are compared to the ones obtained by the standard methods. As demonstrated in numerical examples, the presented **MCMC** formulation of the regularized clustering problem allows to avoid the locality of the obtained minimizers, provides good clustering results even for very ill-posed problems with overlapping clusters and is the computational superior method for long time series.

Keywords: clustering; time series analysis; Markov chain Monte Carlo; non-stationarity; regularization.

[1]**corresponding author**: Free University Berlin, Institute of Mathematics, Arnimallee 6, 14195 Berlin, Germany, jana.dewiljes@math.fu-berlin.de

[2]Courant Institute for Mathematical Sciences, Department of Mathematics and Center for Atmosphere and Ocean Science, New York University, New York, NY 10012-1110, USA, jonjon@cims.nyu.edu

[3]Universita della Svizzera Italiana,Institute of Computational Science, Via Giuseppe Buffi 13, 6900-Lugano, Switzerland, illia.horenko@usi.ch

1 Introduction

Cluster modeling is widely used in many application areas like computational and statistical physics [43, 15], climate/weather research [22, 23, 9, 12, 13, 46, 7] and finance [21, 39, 49]. In context of time series analysis, the aim is usually to detect a hidden process switching between different regimes of a system's behavior, which helps to predict a certain outcome of future events. In most cases the only given information is observation data, which we can regard as a time series. Then the determination of the model and the data-based description of the regime behavior can be formulated as an optimization problem [2, 16]. The main issue thereby is to compute a hidden path, weighting the influence of the data on the various possible cluster models, and, therefore, specifying the transitions between the regimes.

This can be rather difficult since the underlying problem is (i) ill-posed, due to the high number of unknowns in relation to the known parameters, and (ii) the results obtained with a local minimization algorithm depend on the initial parameters, since the corresponding optimization problem is in general non-convex [2].

Therefore, standard formulations of existing cluster modeling methods such as K-Means [26], fuzzy C-Means [2, 18] or Bayesian machine learning approaches (e.g., Gaussian mixture models (**GMMs**) [34, 3] and hidden Markov models (**HMMs**) [11, 5]) do not manage to provide distinguished hidden paths. The negative effect of the ill-posedness on the cluster modeling results is particularly pronounced for overlapping data clusters, i.e., the clusters where affiliations are too difficult to determine since data values can simultaneously be assigned to different clusters [21]. This effect, together with the implicit a priory assumptions that are imposed on the analyzed data by the above-mentioned methods (e.g., stationarity and Markovianity of the hidden/latent path and independence and Gaussianity assumption for the observed process in context of **HMMs**/**GMMs**) may make the obtained results very much dependent on the initialization of the respective numerical scheme (e.g., a choice of the initial parameters in the Expectation-Maximization algorithm for **HMMs**/**GMMs**).

As was demonstrated recently, additional assumptions about certain generalized smoothness of the hidden process can be implemented in context of Tikhonov regularizations [21] or time discrete bounded variational constrains [24]. Both techniques in variational formulation (meaning a minimization of an appropriate functional with respect to a function that is discretized) have been introduced as a clustering framework based on the Finite Element Method (**FEM**) [20, 21, 22, 24]. Compared to Bayesian mixture models like **HMMs** and **GMMs**, the resulting **FEM**-based framework does not rely on the implicit

probabilistic assumptions about the hidden and the observed processes, contains the above standard clustering methods as special cases and can be robustly applied to a much wider class of data and models [36]. Since the Finite Element Method has been deeply studied in the area of PDEs, the **FEM**-clustering framework benefits from the existing theory and numerical PDE-solvers based on finite element discretization in the context of time series analysis. Although the **FEM**-clustering methods can be deployed to determine the persistent states and the transitions of the process even when the data is overlapping, the solution still suffers from the locality of the problem formulation.

Simulated annealing and additional runs with different initial values are commonly proposed in the literature [44, 42] to overcome the locality of an optimization [45]. In the current manuscript we consider a Markov Chain Monte Carlo (**MCMC**) approach to this problem applying the Metropolis algorithm and an adaptive simulating annealing technique. An approach to generate samples from the Boltzmann distribution of a suitable path integral is introduced. Conceptually, the **FEM**-cluster minimization framework becomes the mean field limit of the **MCMC** method in the setting considered here. The major numerical advantage of the presented method compared to the previously introduced Tikhonov-regularized **FEM**-clustering method is that in the **MCMC**-framework it is not necessary to solve a quadratic optimization problem in every iteration step of the algorithm. The necessity to solve a quadratic minimization problem in the $\mathcal{H}^1$-regularized **FEM**-clustering framework [21] represents one of the major computational bottlenecks and slows down the respective numerical algorithm for very long time series.

This paper has the following outline. In Section 2 a brief introduction into cluster modeling theory is presented. In Section 3 the Monte Carlo framework using the Metropolis algorithm is developed and an adaptive numerical annealing algorithm is introduced. The presented framework is illustrated by means of the computational analysis of several generic model examples in Section 4. Section 5 is devoted to the discussion of the obtained results and directions of future investigations.

2 Regularized cluster modeling framework

In the following, we introduce a regularized clustering framework originally proposed in [21]. The key idea is to regularize (e.g., Tikhonov regularization [47]) clustering problems to improve the posedness (in the sense of Hadamard [14]) of the problem formulation. Similar regularizations are frequently employed in image processing [32, 53], statistics (e.g., non-linear regression analysis) [19] and multivariate spline interpolation [51] but are not used in the context of standard

clustering algorithms. In contrast to smooth interpolation problems (for examples see [51]), where the problem formulations are convex (due to the fact that the interpolated quantity is explicitly given), standard clustering problems are non-convex and ill-posed (since the interpolated affiliation quantity weighting the different clusters is not explicitly available). In the context of the **FEM**-clustering approaches [21, 36] the problem is regularized wrt. the persistency of cluster affiliations of data points. Additional information on the regularized clustering technique can also be found in [20, 22, 23, 24, 8].

2.1 Inverse Problem Formulation

Let $X = \{x(0), \ldots, x(T)\}$ be a time-discrete data series with $x(t) \in \mathbb{R}^n$ being the indicated value at time t. The notation $x_t := x(t)$ is used. We assume that the underlying dynamical system can be described by a certain class of mathematical *direct models*

$$x_t = f(\theta(t)), \tag{1}$$

defined by a model function $f(\cdot)$ and a set of (time-dependent) model parameters $\theta(t)$ from some parameter space Ω. Further, it is possible to include a random process in the expression of the model function, e.g.,

$$f(\theta(t)) := \theta(t) + \xi_t, \tag{2}$$

whereas ξ_t is independent identically distributed and has $\mathbb{E}[\xi_t] = 0$ for all t. The random process ξ_t is often interpreted as measurement errors or implicit influences effecting the system. Other examples of this and more general model classes (i.e., classes allowing to describe dynamics with memory, e.g., Markov process) are given in [36].

Solving the problem of finding a suitable time series x_t for given model parameters $\theta(t)$ wrt. the model function f is referred to as *direct mathematical problem*. In this manuscript we consider the opposite problem of finding parameters $\theta(t)$ that describe the dynamical process 'best' by means of the available time series x_t. In order to define the meaning of 'best' in relation to the given data we introduce a *model distance function*:

$$g(x_t, \theta(t)) : \mathbb{R}^n \times \Omega \to \mathbb{R}. \tag{3}$$

A suitable model distance functional 'measures' the distance between the model and the observed time series, thus any metric $d(\cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_0^+$ can be

used to induce an appropriate function g:

$$g(x_t, \theta(t)) := \Big(d(x_t, \mathbb{E}[f(\theta(t))])\Big)^2, \tag{4}$$

whereas the expected value $\mathbb{E}[f(\theta(t))]$ is employed, since the model function f expression is not necessarily deterministic and can include a random term (e.g, see (2)). For the 2-norm (i.e., $d(\Delta, \Psi) = \|\Delta - \psi\|_2$) and considering the model function given in (2) functional g has the analytic expression:

$$g(x_t, \theta(t)) = \|x_t - \theta(t)\|_2^2. \tag{5}$$

An approximation of an optimal (wrt. $g(x_t, \theta(t))$) parameter $\theta^*(t)$ can be obtained solving the following inverse problem

$$L(\theta(t)) = \sum_{t=1}^{\mathbf{T}} g(x_t, \theta(t)) \rightarrow \min_{\theta(t)}. \tag{6}$$

2.2 Interpolation

Unfortunately, the problem formulation given in (6) is ill-posed due to the many unknown parameters in relative relation to the known information. For example the inverse problem

$$L(\theta(t)) = \sum_{t=1}^{\mathbf{T}} \|x_t - \theta(t)\|_2^2 \rightarrow \min_{\theta(t)} \tag{7}$$

has the trivial but meaningless optimal solution $\theta^*(t) := x_t$. To avoid such a meaningless parametrization and in order to directly address the ill-posednes of (6) the regarded system is assumed to be locally stationary. Then the dynamics can be characterized by time-dependent processes $\gamma_i(t)$ describing transitions between the different locally stationary models or *clusters* (characterized by time-independent model parameters θ_i), i.e.,

$$\mathbf{L}(\Theta, \Gamma) = \sum_{t=0}^{T} \sum_{i=1}^{\mathbf{K}} \gamma_i(t) g(x_t, \theta_i) \rightarrow \min_{\Gamma(t), \Theta}. \tag{8}$$

This assumption is not only reasonable but sensible due to the fact that many real life processes evolve much slower than the discrete time steps of the observational data. The hidden process,

$$\Gamma(t) = (\gamma_1(t), \ldots, \gamma_{\mathbf{K}}(t)), \tag{9}$$

weighting the clusters, is referred to as *affiliation vector* and is subject to the following constraints

$$\sum_{i=1}^{\mathbf{K}} \gamma_i(t) = 1 \quad \forall\, t \in [0, T] \tag{10}$$

and

$$\gamma_i(t) \geq 0 \quad \forall\, t, i = 1, \ldots, \mathbf{K}. \tag{11}$$

2.3 Tikhonov-regularization

Although model parameter Γ was introduced assuming that the considered dynamical process, given by the data, changes slowly (i.e., is persistent), it can still exhibit highly non-continuous behavior (i.e., is rapidly jumping between the $\mathbf{K}$ different regimes). Consequently, it is still ill-posed in the sense of Hadamard [14]. To improve the posedness of the problem and to increase the persistency of the process up to a certain degree, we need to add some assumptions concerning Γ. In particular, following [21] we assume, that $\gamma_i(\cdot)$ is weakly differentiable, i.e., $\gamma_i(\cdot)$ is in a path space, embedded in the Sobolev space $\mathcal{H}^1(0, T)$. Using the assumed prior information, we can write (8) in its regularized form:

$$\mathbf{L}^{\epsilon}(\Theta, \Gamma) = \sum_{i=1}^{\mathbf{K}} \left[\sum_{t=0}^{T} \gamma_i(t) g(x_t, \theta_i) + \epsilon^2 \sum_{t=0}^{T-1} (\gamma_i(t+1) - \gamma_i(t))^2 \right] \rightarrow \min_{\Gamma(t), \Theta}. \tag{12}$$

As was demonstrated in [21], this modification of the optimization problem has a smoothing effect on regime transition behavior and 'filters out' all the non-persistent regimes first. For information about Tikhonov-regularization the reader is referred to [47].

An alternative regularization, addressing the ill-posedness of (8), is proposed in [24, 36]. The key idea is to add another constraint which restricts the possible number of transitions. The approach can be motivated regarding functional (8) for a fixed optimal parameter θ^*. Then the optimal process Γ^* is given in form of:

$$\gamma_i^*(t) = \begin{cases} 1 & \text{if } i = \operatorname{argmin}_j \; g(x_t, \theta_j^*), \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

In other words, it is possible to uniquely assign one of the local cluster models θ_i^* to each discrete time step t and to regard the number of jumps between them.

Model parameter Γ is then considered to be subject to the additional constraint

$$||\gamma_i||_{BV} = \sum_{t=0}^{T-1} |\gamma_i(t+1) - \gamma_i(t)| \leqslant \mathbf{C},\ \forall i, \tag{14}$$

i.e., the maximal number of transitions is bounded from above by variable $\mathbf{C}$. Both regularization approaches increase the persistency of the affiliation process Γ. The main difference is that the Tikhonov ansatz leads to a quadratic optimization problem (considering functional $\mathbf{L}(\Theta, \Gamma)$ given in (12) for fixed parameter Θ), whereas the computation of an approximation of an optimal Γ^* with the **BV**-regularization results in a linear optimization problem (details are discussed in the next section).

A direct approach with **MCMC** techniques to the **BV**-regularized version of the optimization problem (8) is hampered by the additional persistency constraint (14). Thus, for the Metropolis algorithm ansatz (see in Section 3), we consider the regularized functional given in (12), i.e., the Tikhonov regularization.

2.4 Numerical approach and computational complexity

Unfortunately, the inverse problem posed in (12) has no general analytic solution and is not convex (i.e., optimization techniques like gradient descent or Newton methods do not necessarily provide a global minimum). Along the lines of [21], we will approach the problem of optimizing (12) employing an algorithm with subspace iterations. The main idea is to exploit that it is possible to determine a local minimum for Θ of $\mathbf{L}^{\epsilon}(\Theta, \Gamma)$ provided that Γ is given and that a local optimum Γ can be computed for a fixed model parameter Θ. In other words, instead of simultaneously finding optimal parameters Γ^* and Θ^* of (12), the problem is split into two subproblems. Iterations over the subspace optimizations ensure a convergence towards local minima Γ and Θ (see [21] for a proof). Since we are interested in global solutions, the subspace algorithm is repeated for different randomly initialized parameters $\Gamma^{(0)}$ and $\Theta^{(0)}$. These additional iterations are considered to be some form of simulated annealing and are commonly used in the context of non-convex optimization problems. [27, 28]. The details of the algorithm are given in the following pseudocode:

Algorithm 1: Subspace algorithm with annealing steps

input : Number of different regimes $\mathbf{K}$, regularization factor ϵ (or for the $\mathbf{BV}$-regularization: transition limit $\mathbf{C}$), number of simulated annealing steps N_{anneal} and optimization tolerance value τol (optional: number of finite element functions $N_{FEM-functions}$)

output: Global optimizers Γ^* and Θ^*

1 $\mathbf{L}_{min} = 1000000$

2 **for** $r = 1 : N_{anneal}$ **do**

3 Generate random initial $\Gamma_r^{[0]}$ and compute $\Theta_r^{[0]}$

4 **while** $|\boldsymbol{L}^{\epsilon}(\Theta_r^{[s]}, \Gamma_r^{[s]}) - \boldsymbol{L}^{\epsilon}(\Theta_r^{[s-1]}, \Gamma_r^{[s-1]})| \geq \tau ol$ **do**

5 *Step 1:*

6 **for** $j = 1 : N_J$ **do**

7 Determine $\Gamma_r^{[s+1]} = \text{argmin } \mathbf{L}^{\epsilon}(\Gamma, \Theta_r^{[s]})$ subject to constraints (10) and (11), whereas $\Theta_r^{[s-1]}$ denotes the current fixed approximation of the optimal Θ^*. Standard techniques for quadratic (Tikhonov regularization) or linear ($\mathbf{BV}$-regularization) optimization problems can be applied, e.g., simplex algorithm [52].

8 *Step 2:*

9 Compute $\Theta_r^{[s+1]} = \text{argmin } \mathbf{L}^{\epsilon}(\Theta, \Gamma_r^{[s+1]})$ for fixed affiliations Γ. This sub-problem strongly depends on the model choice and its computational complexity can range from a simple computation of a deterministic analytic expression (e.g., geometric clustering problem (5)) to quadratic optimization problems with linear equality and inequality constraints (see [8] for examples).

10 $s = s + 1$

11 **if** $\mathbf{L}_{min}^{\epsilon} \geq \mathbf{L}^{\epsilon}(\Gamma_r^*(t), \Theta_r^*)$ **then**

12 $\mathbf{L}_{min}^{\epsilon} = \mathbf{L}^{\epsilon}(\Gamma_r^*(t), \Theta_r^*)$

13 $\Gamma^* = \Gamma_r^*$

14 $\Theta^* = \Theta_r^*$

15 Return Γ^* and Θ^*

To determine Γ in Step 1 (see lines 5-7) of the subspace algorithm with simulated annealing iterations one needs to solve a quadratic optimization problem with linear constraints. Such problems are known to be $\mathbf{NP}$-complete [50] and are, therefore, considerably expensive regarding the run time. It is possible, however, to reduce the dimension of Γ and, therefore, the computational complexity by using ideas from the Finite Element Method ($\mathbf{FEM}$). The key idea is to discretize the process Γ with a number of $N_{FEM-functions}$ finite element functions

and to continue optimizing the reduced problem wrt. the discretized $\tilde{\Gamma}$. Then the main advantage is that the run time of Step 1 (depending on the size of the affiliation process) can be considerably decreased (especially for very persistent dynamical systems where the number of required finite element functions to obtain a qualitative solution is much smaller than the time dimension, i.e., $N_{FEM-functions} \ll T$).We refer to the proposed framework as **FEM**-clustering technique.

In case the **BV**-regularization is employed to create persistency in the time interval, the problem to find an optimal Γ^* subject to the additional constraint (14) becomes a linear optimization problem with constraints. This problem is also known to have exponential run time in the worst case (see simplex algorithm [52]). We refer to this ansatz as **FEM-BV**-clustering framework.

An **MCMC** approach has only linear complexity, the details of which will be discussed in the Section 3. A run time comparison (of the **MCMC**- and the **FEM-BV**-clustering method) on the basis of synthetic high dimensional data is considered in Section 4.

Step 2 (see lines 8-9) of the subspace algorithm on the other hand depends on the choice of the underlying model class (1). In the following we will consider the example model function (2) with model distance function (5). Then the computation of optimal parameters θ_i^* for fixed optimal affiliations Γ^* conforms to

$$\theta_i^* := \frac{\sum_{t=0}^{T} \gamma_i(t) x_t}{\sum_{t=0}^{T} \gamma_i(t)}. \tag{15}$$

This deterministic analytic expression for a minimal θ^* reduces the optimization problem (12) to

$$\overline{\mathbf{L}}^{\epsilon}(\Gamma) = \sum_{i=1}^{\mathbf{K}} \left[\sum_{t=0}^{T} \gamma_i(t) \|x_t - \frac{\sum_{t=0}^{T} \gamma_i(t) x_t}{\sum_{t=0}^{T} \gamma_i(t)}\|_2^2 + \epsilon^2 \sum_{t=0}^{T-1} (\gamma_i(t+1) - \gamma_i(t))^2 \right] \to \min_{\Gamma(t)} \tag{16}$$

with the conditions (10) and (11).

2.5 Information criterion

The choice of an optimal **K** wrt. the system given by the data x_t presents another challenge. This problem has already been discussed in context of the **FEM**-clustering framework and the interested reader is referred to [21] and [23], where some ways of choosing an optimal **K** are presented. While aiming at selecting the best possible model, we also want to avoid over-fitting. Thus, we make use of an *information criterion* (e.g., Akaike [1] or Bayesian [48]) to find an optimal **K**. The conceptual idea of an information criterion is to regard the

balance of the quality and the complexity of a computed model. For example, the structure of a standard Bayesian information criterion (**BIC**) can be adapted to fit the proposed clustering problem (12):

$$\mathbf{BIC}(\mathbf{K}) = -2\log(\mathcal{L}(\mathbf{K})) + |M(\mathbf{K})|\log(T). \tag{17}$$

In detail this means that the log-likelihood function

$$\mathcal{L}(\mathbf{K}) = \prod_{t=1}^{T}\sum_{i=1}^{\mathbf{K}}\gamma_i(t)\rho_i\Big(g(x_t,\theta_i);\Lambda_i\Big) \tag{18}$$

of functions ρ_i (which are fitted to the residual processes $g(x_t,\theta_i)$ corresponding to model parameters θ_i for $\mathbf{K}$ different local models) is weighed against the total number of involved parameters $|\mathrm{M}(\mathbf{K})|$. A detailed derivation of the likelihood function is discussed in [36] in the context of a modified version of Akaike's information criterion (the only difference to the Bayesian information criterion is that $|\mathrm{M}(\mathbf{K})|$ is not multiplied with the logarithm of the number of time steps T) for the proposed clustering problem. Regarding the geometric model proposed in (5), the total number of involved parameters is:

$$|M_{geometic}(\mathbf{K})| := \left(\sum_{i=1}^{\mathbf{K}-1}\sum_{t=0}^{T-1}|\gamma_i(t+1)-\gamma_i(t)|\right)\mathbf{K}n \tag{19}$$

with n being the dimension of model parameter vectors θ_i. In case the model parameters are determined employing the **FEM-BV**-clustering framework the likelihood does not only depend on the number of different regimes $\mathbf{K}$ but also on the maximum number of transitions $\mathbf{C}$.

3 Deploying MCMC methods

The main computational drawback concerning the average clustering function proposed in (16) arises from the fact that $\overline{\mathbf{L}}^{\epsilon}(\Gamma)$ is non-convex (note that $\mathbf{L}^{\epsilon}(\Theta,\Gamma)$ given in (12) in general is also non-convex). To evade the locality of the numerical solution and directly obtain a global solution, a variational approach in form of the following probabilistic formulation

$$\pi_{\overline{\mathbf{L}}^{\epsilon},\beta}(\Gamma) = \frac{1}{Z}\exp(-\beta\overline{\mathbf{L}}^{\epsilon}(\Gamma)) \tag{20}$$

with

$$Z = \int_{\Gamma}\exp(-\beta\overline{\mathbf{L}}^{\epsilon}(\Gamma)), \tag{21}$$

called the Boltzmann distribution, is considered. An optimal parameter Γ^*, minimizing (16), can be approximated by sampling from the above distribution (20). Whereas $\overline{\mathbf{L}}^{\epsilon}(\Gamma)$ with fixed ϵ is referred to as *energy function* and $\beta > 0$ is a real variable named *inverse temperature parameter*. The Boltzmann distribution has its origin in statistical physics, where it describes the probability of a particle's speed, depending on the temperature of a system. Z is a *normalizing constant*, ensuring that $\int_{\Gamma} \pi_{\overline{\mathbf{L}}^{\epsilon},\beta}(\Gamma) = 1$. Boltzmann distributed samples have the property to be forced towards the minimal energy configuration as β is tending to ∞, meaning that the probability to obtain samples that minimize $\overline{\mathbf{L}}^{\epsilon}(\cdot)$ grows as β increases. Therefore, the solution of the optimization problem (16) can be approximated by generating samples that are Boltzmann distributed. However, computing a normalizing constant such as Z is difficult since it implies a numerical calculation of the integral (21) for many dimensions. For problems like that, the Metropolis algorithm [6, 17, 35] is a useful tool, due to the fact that it does not require to determine the normalization constant Z.

The underlying principle of this **MCMC** framework is to generate a Markov chain of samples, having a certain *target distribution* (e.g., $\pi_{\overline{\mathbf{L}}^{\epsilon},\beta}$) as its unique stationary distribution. The construction of the Markov chain requires to choose a density $q(\cdot,\cdot)$, referred to as *proposal density*, which is used to propose the next possible element of the chain. A proposed sample is either accepted to be an element of the chain or not. The acceptance-rejection sampling takes place in form of

$$\alpha(\Gamma,\Gamma') = \begin{cases} \min\left\{1, \frac{\pi_{\overline{\mathbf{L}}^{\epsilon},\beta}(\Gamma')q(\Gamma',\Gamma)}{\pi_{\overline{\mathbf{L}}^{\epsilon},\beta}(\Gamma)q(\Gamma,\Gamma')}\right\} & \text{if } \pi_{\overline{\mathbf{L}}^{\epsilon},\beta}(\Gamma)q(\Gamma,\Gamma') > 0. \\ 1 & \text{otherwise} \end{cases} \tag{22}$$

In the following we will consider a *random walk* family of densities $q(\cdot,\cdot)$, i.e., a new proposal depends only on a random *noise* η. Random Walk Metropolis (RWM) has, contrary to other proposal density families (e.g., independent sampling [31] which has the best performance for proposal densities similar to the target distribution), the advantage not to require any additional knowledge about the target distribution $\pi_{\overline{\mathbf{L}}^{\epsilon},\beta}$. It is also possible to include gradient information for the proposal of a new potential sample of the Markov chain (e.g., Metropolis Adjusted Langevin Algorithm (MALA) [40, 38]) which is hampered by the fact that the gradient might not exist for the proposed problem or that the gradient may not be bounded (due to the constraints).

In this manuscript the Gaussian proposal density

$$q(\Gamma, \Gamma') = q(\Gamma' - \Gamma) = q(\eta) = \frac{1}{(2\pi)^{\frac{T}{2}} |\Sigma|^{\frac{1}{2}}} \exp\Big(-\frac{1}{2}(\eta - \mu)^\top \Sigma^{-1}(\eta - \mu)\Big) \quad (23)$$

is employed to generate the noise η, which is added to the current element Γ to obtain a new candidate

$$\Gamma' = \Gamma + \eta \quad (24)$$

with expected value $\mu_\eta = \mathbf{0}$ and identity covariance matrix Σ_η. Wether a proposed sample Γ' is going to be accepted strongly depends on the variance of the noise η. The energy $\overline{\mathbf{L}}^\epsilon(\Gamma')$ is likely to be similar to the energy of the last element of the chain if the variance of η is small. On the other hand, it is important to propose samples which differ from the previous chain member so that the entire sample space is traversed. This, however, can be achieved when the noise η has a relatively large variance. Concluding, it is necessary to gain some control over η by adjusting its variance. Instead of directly changing the covariance matrix Σ_η, a new variable ν is added to the random walk equation (24)

$$\Gamma' = \Gamma + \nu\eta \quad (25)$$

which we will refer to as *noise factor*. Due to the underlying normal distribution (23) the additional variable ν is adjusting the variance of η by a factor of ν^2. The proposed Metropolis algorithm, used for the clustering problem (16), is

given in the following pseudocode.

Algorithm 2: **MCMC**-clustering approach

input : Number of different regimes $\mathbf{K}$, regularization factor ϵ, length of Markov chain $N_{MC-length}$, ν to adjust variance of random walk noise η and cooling schedule $\{\beta^{(0)}, \dots, \beta^{(N_{MC-length})}\}$ (optional: number of finite element functions $N_{FEM-functions}$)

output: Global optimizer Γ^*

Choose or generate an initial value $\Gamma^{(0)}$ (e.g., uniform initial distribution)
for $r = 1 : N_{MC-length}$ **do**
    Generate $\eta \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ and u from $\mathcal{U}(0,1)$, the uniform distribution and propose a new sample $\Gamma' = \Gamma^{(r-1)} + \nu\eta$ (subject to the constraints (10) and (11), details are discussed in the next section).
    Calculate the acceptance rate $\alpha(\Gamma^{(r-1)}, \Gamma')$ given in (22) with $\beta^{(r)}$
    if $u \leqslant \alpha(\Gamma^{(r-1)}, \Gamma')$ **then**
        set $\Gamma^{(r)} = \Gamma'$
    else
        set $\Gamma^{(r)} = \Gamma^{(r-1)}$
Return $\Gamma^{(N_{MC-length})}$

In order to sample a $\pi_{\overline{\mathbf{L}}^{\epsilon},\beta}$ distributed Markov chain $\Gamma^{(0)}, \dots, \Gamma^{(N_{MC-length})}$, another simulated annealing technique is employed. The concept of simulated annealing in this context allows us to obtain samples $\Gamma^{(s)}$ with smaller energy $\overline{\mathbf{L}}^{\epsilon}(\Gamma^{(s)})$ by slowly reducing the current temperature $\beta^{(s)}$, hence the term *annealing*. Although this methods improves the results, the general disadvantages are very slow convergence and the fact that it is not possible to determine whether an optimal solution has been obtained. An adaptive simulated annealing scheme, where the cooling schedule depends on convergence diagnostics [6, 41], is proposed in Section 4. Various other techniques, approaching the drawbacks of simulated annealing, have been introduced in recent years, the interested reader is referred to methods such as simulated sinstering [30], simulated tempering [33], and sequential Monte Carlo [37]. The Metropolis approach has a linear computational complexity, i.e., $\mathcal{O}(\mathbf{K}(N_{FEM-functions})N_{MC-length})$. A direct complexity comparison of the Metropolis- and the **FEM**-clustering (which is a NP-complete [50] numerical problem) framework, reveal that the **MCMC** approach has in general a smaller run time. An exemplary run time comparison of the Metropolis algorithm and the **FEM-BV**-clustering, by means of a multidimensional synthetic time series, is displayed in Section 4.4.

The regarded clustering problem (16) has the advantage that the optimization only depends on Γ due to the fact that optimal model parameters θ_i^* can

be uniquely determined for fixed affiliations Γ. In general, however, considering arbitrary model functions f (for different examples see [36]), there is no analytic expression for the parameters θ_i. Nevertheless, the Metropolis algorithm can still be employed for the sub-problem (12) for fixed Θ (see Step 1 in pseudocode in Section 2.4) and, depending on the model choice (1), it is also possible to determine model parameters θ_i for fixed Γ (see Step 2 in pseudocode Section 2.4). In other words, the subspace algorithm proposed in Section 2.4 can be used in combination with the Metropolis algorithm to tackle other model function class problems where the optimization can not be reduced to one model parameter.

3.1 Constraints on Γ

It is important to point out that Γ still has to satisfy the conditions (10) and (11). In the special case $\mathbf{K} = 2$, we sample the path $\gamma_1(\cdot)$ with $\gamma_1(t) \in [0, 1]$ (i.e., the proposed sample Γ' is modified to suit the boundaries by setting the entries of Γ' greater than one to one and the negative values to zero) and by means of (10) it is possible to obtain

$$\gamma_2(\cdot) = 1 - \gamma_1(\cdot). \tag{26}$$

To generate an affiliation vector with an arbitrary number of relating cluster models θ_i, i.e., not being limited to $\mathbf{K} = 2$, one of the possibilities is to assume

$$\gamma_i(t) = \frac{\exp(a_i(t))}{\sum_{j=1}^{\mathbf{K}} \exp(a_j(t))} \tag{27}$$

and sample with respect to $a_i(t) \in \mathbb{R}$. The choice of this analytic expression ensures that the constraints (10) and (11) are automatically fulfilled.

4 Numerical Examples

We presented an **MCMC** approach to regularized clustering optimization problems. Now we want to investigate the proposed method by applying it to several sets of generic model data, which vary in size and type.

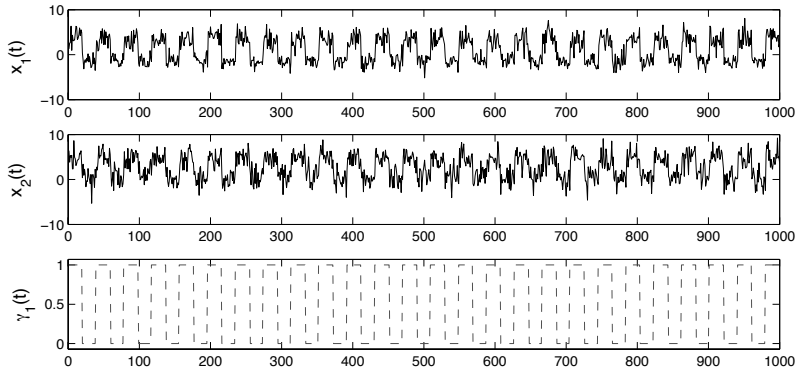

Figure 1: *The two-dimensional time series x_t (upper and center panel) switches between two multivariate normal distributions with different expected values μ_i (28), and identity covariance matrixes Σ_i. In the lower panel, the corresponding hidden process, switching between the two distributions, is shown.*

First, we apply the algorithm to the synthetic data, designed by means of the predefined cluster switching process, shown in the lower panel of Figure 1. The two local stationary models θ_1 and θ_2 are chosen to be multivariate normal distributions, given by the expected values

$$\mu_{\mathbf{1}}(\mathbf{t}) = \begin{pmatrix} 3.5 \\ 5 \end{pmatrix}, \mu_{\mathbf{2}}(\mathbf{t}) = \begin{pmatrix} -1.5 \\ 0.5 \end{pmatrix}, \tag{28}$$

and with the identity as covariance matrixes Σ_i (Figure 1 upper and center panel). Before we can compare the **FEM**-clustering methodology [21, 20, 22, 23] with the **MCMC**-clustering methods presented above, we have to consider the cooling schedule and the noise factor ν utilized in the **MCMC** algorithm (see pseudocode in Section 3).

4.1 Technique choices and parameter scaling

In this section an insight into the scaling, necessary to obtain good approximations of an argument Γ minimizing the energy function (16), is provided. As already discussed above, a Random Walk Metropolis (RWM) algorithm with Gaussian noise $\eta \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$[1] is employed. It is possible to change the variance of η with a factor ν (see (25)) to influence the acceptance-rejection-procedure given in (22). Moreover, the cooling schedule is addressed. As was already pointed out, a cooling schedule ensuring a high probability to approach a minimal argument of the energy function (16), is not practical due to very slow

[1]The required pseudorandom numbers are generated using the *Mersenne Twister* (see http://www.math.sci.hiroshima-u.ac.jp/ m-mat/MT/emt.html)

convergence. Instead of this we will, depending on the relation between the total number of accepted samples and those actually having a smaller energy function value than the previous sample, gradually update the inverse temperature parameter β. This choice gives us the option to influence the acceptance process during the run of the **MCMC** algorithm and β is only increased whenever the energy of the samples in a certain time interval is jumping up too much rather than going down.

Before the adaptive updated scheme is proposed for the variables ν and β, we will discuss their particular influence on the acceptance rate and the energy. Moreover, the impact of the regularization factor ϵ is considered. Results of different **MCMC** runs, each concentrating on one of the variables, are displayed in Table 1. The Metropolis algorithm is applied to the time series given in Figure 1 and produces Markov chains of length 100000. Further, the displayed **MCMC** energy values and acceptance rate percentages are the means over 100 runs of the Metropolis algorithm. The parameters β and ν are fixed during the run of the method, since, in this particular setting, it is easier to investigate the effects of the different values. The corresponding results for the hidden process, determined by the **FEM**-clustering method [20, 21, 22, 23], (specifications: $N_{FEM-functions} = 100$, $N_{anneal} = 10$ and $\tau ol = 0.0000001$), and the energy of the synthetic process (Figure 1, lower panel) are given.
The upper panel of Table 1 reveals the behavior of the energy with respect to the changing parameter β. It becomes apparent that a higher value of β leads to a setting, where a proposed element is only accepted if it has a similar or lower energy than the current element of the Markov chain. This implies that the acceptance rate decreases, since less movement within the system is permitted, which hampers reducing the energy of the samples. This effect can be seen best regarding the results of the Metropolis algorithm with $\beta = 1000$ and $\beta = 1.0E + 09$. Nevertheless, the value for the inverse temperature should not be chosen too small, as the acceptance rate of nearly 80% and the corresponding high energy for $\beta = 1$ demonstrate. Consequently, it is difficult to achieve that the entire sample space is traversed and at the same time permit a too high acceptance rate.

The center panel of the Table 1 illustrates the smoothing effect influencing the energy, caused by the regularization factor ϵ. A growing ϵ leads to higher energy values, since in the energy function formula (16) the regularization summand of $\overline{\mathbf{L}}^{\epsilon}(\cdot)$ is multiplied with the square of ϵ. This fact can explain comparatively low energy values for small ϵ. On the other hand, a higher regularization factor smoothes the transition process Γ, meaning that short transitions in the process are evened out and the resulting cluster states become more persistent. However, the parameter value should not be too large, since then the regular-

β	noise factor	ϵ	acceptance rate	$\overline{\mathbf{L}}^{\epsilon}(\Gamma_{MCMC})$	$\overline{\mathbf{L}}^{\epsilon}(\Gamma_{FEM})$	$\overline{\mathbf{L}}^{\epsilon}(\Gamma_{syn})$
1	0.001	5	78.80 %	15356.0	11702.0	7832.9
10	0.001	5	35.82 %	7079.4	11702.0	7832.9
1000	0.001	5	29.14 %	7033.1	11702.0	7832.9
1000000	0.001	5	29.09 %	7089.0	11702.0	7832.9
1.00e+09	0.001	5	28.98 %	7107.4	11702.0	7832.9

ϵ	β	noise factor	acceptance rate	$\overline{\mathbf{L}}^{\epsilon}(\Gamma_{MCMC})$	$\overline{\mathbf{L}}^{\epsilon}(\Gamma_{FEM})$	$\overline{\mathbf{L}}^{\epsilon}(\Gamma_{syn})$
0	1.00e+09	0.001	21.24 %	6858.5	9078.4	5332.9
2	1.00e+09	0.001	25.48 %	6138.0	9472.8	5732.9
5	1.00e+09	0.001	28.98 %	7107.4	11702.0	7832.9
12	1.00e+09	0.001	13.78 %	15482.0	15498.0	19733.0

noise factor	β	ϵ	acceptance rate	$\overline{\mathbf{L}}^{\epsilon}(\Gamma_{MCMC})$	$\overline{\mathbf{L}}^{\epsilon}(\Gamma_{FEM})$	$\overline{\mathbf{L}}^{\epsilon}(\Gamma_{syn})$
0.1	1.00e+09	5	0.61 %	15554.0	11702.0	7832.9
0.01	1.00e+09	5	2.9 %	9178.9	11702.0	7832.9
0.001	1.00e+09	5	28.98 %	7107.4	11702.0	7832.9
0.0001	1.00e+09	5	49.74 %	15524.0	11702.0	7832.9

Table 1: *The three panels display the data of the* ***MCMC*** *methodology runs by means of the time series, shown in Figure 1, with different values for the parameters β and ϵ and the noise factor ν. Instead of a cooling schedule, we fix β to sample from the Boltzmann distribution and the possible elements of the Markov chain are proposed by a normal density with a fixed noise factor. These measures are taken to motivate the influences of the variables on the quality of the resulting approximation of the hidden process, which is rated by its energy value, shown in the tables. Each of the results of the* ***MCMC*** *method is the mean of* 100 *different runs. To create equal conditions for the Metropolis algorithm runs, we set a uniformly distributed initial γ.*

ization factor is dominating the energy (see row with $\epsilon = 12$). If, however, the particular parameter is chosen carefully, it is possible to obtain a very good approximation of a minimal Γ^*.
The noise factor ν influences the acceptance rate of the Metropolis algorithm significantly, as shown in the lower panel of Table 1. The considerable difference between the acceptance rate for $\nu = 0.1$ and $\nu = 0.0001$ demonstrates that the noise factor needs accurate adjustment. The best energy value can be obtained for $\nu = 0.001$ due to an acceptance rate of 28.98%, which is close to the supposedly optimal percentage proposed in [6].

	0	10	20	30	40	50	60	70	80	90	end
ν	0.1	0.085	0.03	0.005	0.002	0.001	6e-4	3e-4	2.4e-4	1.7e-4	1.3e-4
β	1	1	1.1	2.08	5.9	15.4	23.4	23.4	74.8	102.6	237.1

Table 2: *The table shows the development of the parameter β and the noise factor ν and their updates during the run of the Metropolis algorithm, applied to the data from Figure 1. The algorithm is set with $\epsilon = 5$, $\beta = 1$, initial noise factor $\nu = 0.1$ and a Markov chain of length 100000. The updates of the inverse temperature are made in steps of 1000 and β is increased if the number of accepted samples with a higher energy than the previous element of the chain is more than 10%. The lower row of the table demonstrates the change that the noise factor ν undergoes during the adaption process of the **MCMC** method. The adjustment of the parameter ν is done on the basis of the data of the previous 500 iterations every 1000 steps of the method, however shifted (by* 500 *steps) to the one of β.*

Concluding, the adjustable sampling parameters ν, β and ϵ have a considerable effect on the performance of the **MCMC** methodology. Contrary to the regularization factor ϵ, the variables ν and β require careful adjustment during the run of the Metropolis algorithm, since different types of influences are needed in the different stages of the method.

In the following we introduce the adaptive update scheme used to tune the variables β and ν during the sampling process. The framework is motivated by the reference value 23.4% which is theoretically verified to be the optimal (concerning the results) acceptance rate [6, 41]. As already discussed, the inverse temperature should be increased, however, not too fast, since otherwise samples of a region in the sample space, different to the one of the current element of the chain, are rarely going to be accepted. Therefore, β is updated after 1000 proposals of the run depending on the ratio between the number of accepted samples with a lower energy than the previous element of the chain and the total number of accepted samples (for details see pseudocode below).

The movement during one run of the **MCMC** method with an initial value $\beta = 1$ is shown in Table 2. It is apparent that the inverse temperature value is raised very slowly till the chain already has a length of 60000 and then rapidly grows. A similar adaption process is used to optimize the noise factor develop-

ment (Table 2).

The parameter ν is also adjusted every 1000 steps, however, shifted (by 500 steps) to the updates of β to avoid too extreme changes. The noise factor is either increased or reduced, depending on the percentage of accepted samples, which is supposed to be around 23.4%. The algorithm starts with an initial value $\nu = 0.1$ and the panel of Table 2 shows that the noise factor is decreasing very fast, meaning that not enough proposed samples are accepted. In order to summarize the proposed procedure, the pseudocode of the Metropolis algorithm is considered again, concentrating only on the adaptive annealing scheme [2] for

[2] A cpp implementation of the algorithm with simulated annealing scheme is available on `http://www.dewiljes.de/dewiljes/Jana.html`

the inverse temperature β and the update of the variable ν.

Algorithm 3: Adaptive β and ν update

input : Number of different regimes $\mathbf{K}$, value for regularization factor ϵ, length of Markov chain $N_{MC-length}$, initial values for the noise factor ν and inverse temperature β (optional: number of finite element functions $N_{FEM-functions}$)

output: Global optimizer Γ^*

1 Choose or generate an initial $\Gamma^{(0)}$, $\beta^{(0)}$ and $\nu^{(0)}$.
2 **for** $r = 1 : N_{MC-length}$ **do**
3 Propose new sample Γ'
4 Accept/Reject -procedure
5 **if** *accept* **then**
6 $N_{accept} = N_{accept} + 1$
7 **if** $\overline{L}_\epsilon(\Gamma^{(r-1)}) > \overline{L}_\epsilon(\Gamma')$ **then**
8 $N_{accept-lowerEnergy} = N_{accept-lowerEnergy} + 1$
9 **if** mod $(r, 1000) = 0$ **then**
10 **if** $N_{accept} < 90$ **then**
11 $\nu = \nu \cdot 0.85$
12 **else if** $N_{accept} > 140$ **then**
13 $\nu = \nu \cdot 1.05$
14 **else**
15 $\nu = \nu$
16 $N_{accept} = 0$
17 $N_{accept-lowerEnergy} = 0$
18 **if** mod $(r, 1000) = 500$ **then**
19 **if** $N_{accept} - N_{accept-lowerEnergy} \geq N_{accept} \cdot 0.25$ **then**
20 $\beta = \beta \cdot 1.111$;
21 $N_{accept} = 0$
22 $N_{accept-lowerEnergy} = 0$
23 Return $\Gamma^{(N_{MC-length})}$

For the update of the noise factor ν in lines 9-15 we regard the interval of 18% (see line 10: $0.18 = \frac{90}{500}$) to 28% (see line 12: $0.28 = \frac{140}{500}$) surrounding the optimal 23.4% of the accepted samples in relation to the overall proposed samples. If the number of accepted samples (only considering the past 500 steps) is outside the regarded interval of percentages, we adaptively change the noise factor (see lines 11 and 13). The inverse temperature β is increased (see

lines 18-20) if less than 75% of the accepted samples have lower energy, i.e.,

$$\frac{N_{accept} - N_{accept-lowerEnergy}}{N_{accept}} \geq 0.25. \tag{29}$$

These dynamic changes of the two parameters β and ν improve the quality of the results of the Metropolis application immensely. In the following paragraph we will continue to investigate the **MCMC** method and compare it to the variational **FEM**-clustering approach [21].

4.2 Comparison of MCMC and FEM-applications

In the previous paragraph, an update function for the noise factor ν and an adaptive method to increase the parameter β, which acts as an amended version of simulated annealing, were proposed. These settings of the Metropolis algorithm are used for all the test cases in this section. Firstly, we consider the influence of the regularization factor ϵ again and demonstrate its smoothing effect via the four graphic panels of Figure 2, each displaying an approximation of the optimal Γ^* calculated with different ϵ values. The impact of ϵ on the results of the Metropolis algorithm is illustrated in Figure 2. It is apparent that the cluster classification becomes more distinctive with a growing ϵ value. However, if the ϵ value is too high, the graph approaches the middle line between the models (Figure 2, $\epsilon = 12$), which makes it impossible to relate the persistent states to the corresponding cluster. Besides establishing the influence of the variable ϵ in Figure 2, we want to draw a comparison between the **FEM**-clustering methodology and the **MCMC** application. Therefore, the graphs of Figure 3 display approximations of the hidden process Γ^*, calculated with the **FEM**-clustering algorithm for $\epsilon = 0, 2, 5, 12$.

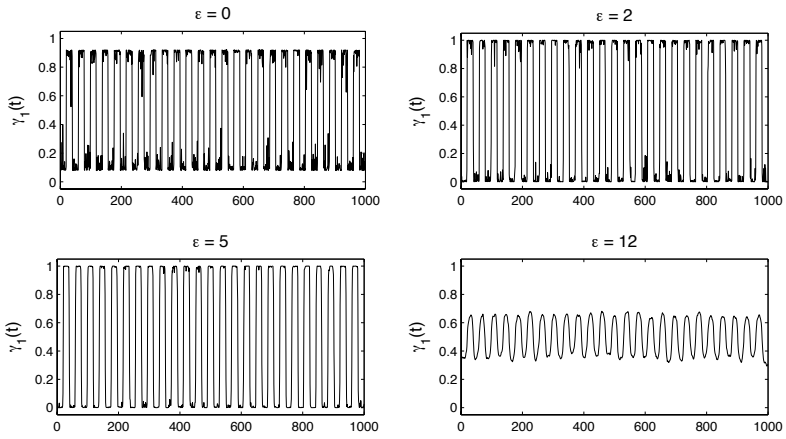

Figure 2: *The four graphic panels display approximations of the hidden process* Γ^**, obtained by the* ***MCMC*** *method on the basis of the time series of Figure 1. We demonstrate the smoothing effect of four different regularization parameter values* $\epsilon =$ *0,2,5,12.*

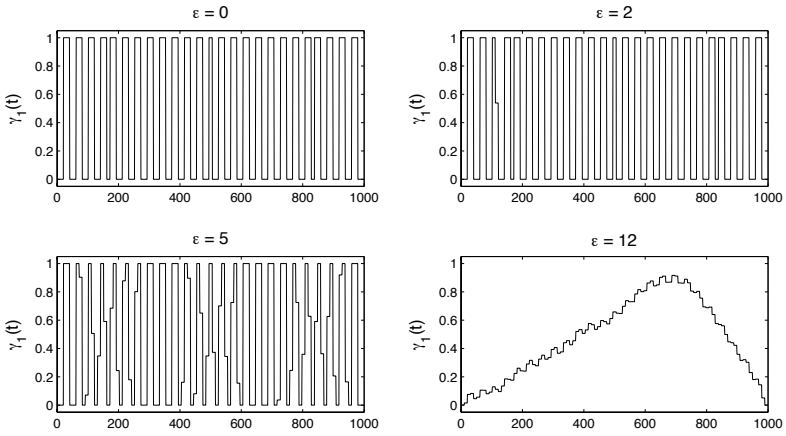

Figure 3: *The four graphs serve as reference and comparison to the* ***MCMC*** *of Figure 2. Here the hidden paths for a selection of four regularization factor values (*$\epsilon =$ *0,2,5,12), determined with the* ***FEM****-clustering algorithm (*$N_{FEM-functions} = 100$*,* $N_{anneal} = 10$ *and* $\tau ol = 0.0000001$*), are shown.*

Firstly, it is conspicuous that the sensitivity of the **FEM**-clustering results, regarding the ϵ parameter, is different to the reaction of Metropolis algorithm concerning the changes of the regularization factor. In other words, note that the optimal ϵ value for the **MCMC** technique might lead to very bad results for the **FEM**-methodology. However, both algorithms react strongly to a high regularization factor (fourth panel of Figures 2 and 3). Moreover, selecting an optimal

ϵ	initial ν	initial β	acceptance rate	$\overline{\mathbf{L}}^{\epsilon}(\Gamma_{MCMC})$	$\overline{\mathbf{L}}^{\epsilon}(\Gamma_{FEM})$	$\overline{\mathbf{L}}^{\epsilon}(\Gamma_{syn})$
0	0.1	1	16.35 %	8554.4	9078.4	5332.9
2	0.1	1	16.42 %	5376.1	9472.8	5732.9
5	0.1	1	16.65 %	6545.2	11702.0	7832.9
12	0.1	1	red18.69 %	15024.0	15498.0	19733.0

Table 3: *The data, recorded in the table, corresponds to the graphic panels of Figures 2 and 3.*

ϵ value does not only depend on the technique of choice, but also on the type of the time series the algorithm is applied to. Then it is important to note that the process, determined within the **FEM**-clustering framework, is more definite than the one, obtained with the **MCMC** method. This can be explained with the number of chosen finite element functions ($N_{FEM-functions} = 100$), which automatically leads to a more distinguished path. However, the energy values of the results of the Metropolis algorithm are much smaller than the ones of the **FEM**-clustering technique as can be seen in Table 3, which displays the corresponding data of the graphs of Figure 2. The improvements made, concerning the adaptive adjusting of β and ν, clearly are reflected in the energy values of Table 3, which we were able to reduce significantly, regarding the former results of Table 1. Furthermore, the acceptance rate displayed for $\epsilon = 5$ supports the conclusion drawn from the noise factor development (Table 2), where we stated that the value for ν is decreasing, because the acceptance rate is below the value, typically used in the literature [6, 41].

It is important to mention that a comparison in terms of the energy function (16) represents one of the most conceptually adequate quality measures of the performance of clustering methods. Most of the existing clustering approaches can be formulated as optimization problems with respect to an appropriate energy function. For example the K-Means method can be understood as an iterative minimization of energy function (12) for $\epsilon = 0$. Although the value of the energy function is a good evaluation tool when it comes to choosing optimal tuning parameters or comparing different techniques, the approximation of the hidden process $\gamma_i(t)$ relating to the smallest energy is not necessarily the best model for the considered time series. In case of synthetic data we can compare the θ_i with the expected values μ_i to investigate the quality of the obtained models. We calculate the model parameters for the results of the **FEM**-clustering technique and the **MCMC** method with $\epsilon = 5$:

$$\theta_{\mathbf{1}}^{\mathbf{MCMC}}(\mathbf{t}) = \begin{pmatrix} 3.5310 \\ 4.8928 \end{pmatrix}, \theta_{\mathbf{2}}^{\mathbf{MCMC}}(\mathbf{t}) = \begin{pmatrix} -1.4864 \\ 0.4391 \end{pmatrix} \tag{30}$$

and

$$\theta_{\mathbf{1}}^{\mathbf{FEM}}(\mathbf{t}) = \begin{pmatrix} 2.9619 \\ 4.3771 \end{pmatrix}, \theta_{\mathbf{2}}^{\mathbf{FEM}}(\mathbf{t}) = \begin{pmatrix} -0.8400 \\ 1.0237 \end{pmatrix}. \tag{31}$$

The values obtained from the **MCMC**-clustering approach differ only a little from the mean values (given in (28)) used to generate the synthetic time series. The model parameters determined with the **FEM**-clustering algorithm, however, show a larger deviation.

It is possible to evaluate the quality of the results by reproducing the time series via the model parameters θ_i and the corresponding affiliation vectors $\gamma_i(\cdot)$ and compare it with the actual data. Figure 4 displays histograms of the difference between the reproduced and the real data for the **MCMC** method and the **FEM**-clustering technique.

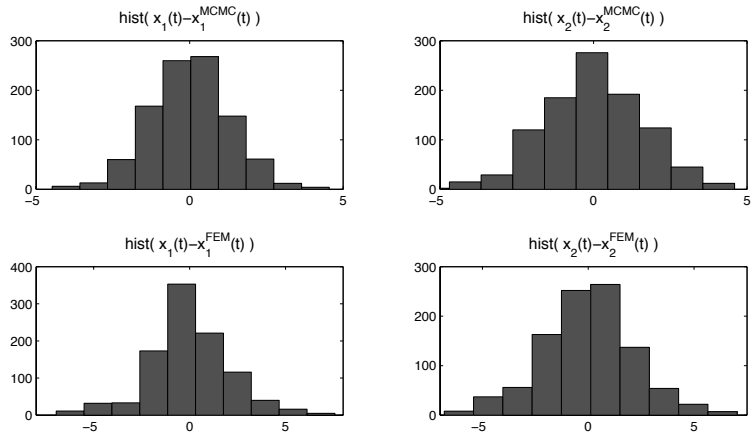

Figure 4: *The difference between the two-dimensional time series x_t and the reproduced data is shown in form of a histogram for the **FEM**-clustering framework and the **MCMC** algorithm. The **MCMC** deviation frequencies for the first and second coordinate are displayed in the upper panels and the results relating to the **FEM**-clustering in the lower panels of the figure.*

The histograms, shown in the upper two panels of Figure 4, resemble normal distributions with standard deviation one and mean value zero. Meaning that the approximated model parameters θ_i and the relating cluster weighting functions $\gamma_i(\cdot)$, obtained with the **MCMC**-clustering framework, are a good characterization of the regime corresponding to the data. The lower panels display histograms with a larger variance but still of normal distributed nature.

The next paragraph will deal with the behavior of the introduced **MCMC** method applied to overlapping time series.

4.3 Robustness concerning overlapping distributions

In contrast to the data we have considered so far, where the transitions have been clearly visible, we now want to examine the **MCMC** framework under the conditions that the expected values of the distributions, the data is generated from, are 'coming closer' together. The following three figures demonstrate the influence of overlapping data in form of a time series on the resulting affiliation vector $\gamma_i(t)$ given by the Metropolis algorithm.

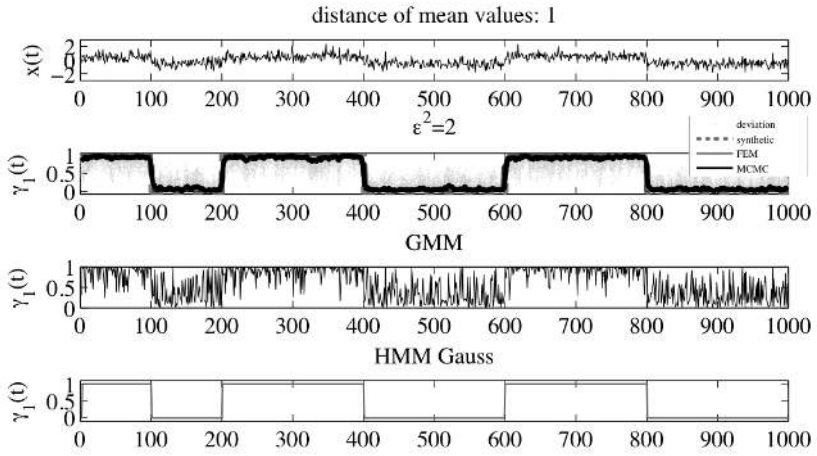

Figure 5: *The upper panel displays a synthetic time series generated with random values of two normal distributions with a distance of 2 between the expected values μ_i ($\sigma_i = 0.5$). The paths identifying the switching between the possible two clusters, computed with either the **MCMC** method ($\epsilon^2 = 2$, length of chain 100000, 100 runs) and the **FEM**-clustering framework, are shown in the second panel from the top. Moreover, the graph includes the synthetic process and the results of the 100 metropolis runs, used to calculate the mean, are shown in form of grey dots. The process displayed in the third panel of the figure is obtained with **GMMs** and the one in the fourth panel with **HMM**. Both serve as a comparison.*

We make use of two normal distributions with equal variances $\sigma_i = 0.5$ to create one-dimensional time series. Furthermore, we restrict the processes to a number of five transitions so that the impact of overlapping data can be demonstrated easier. The graphs include the hidden processes, calculated with the version of the **FEM**-clustering algorithm (numerical specifications: $N_{FEM-functions} = 100$, $N_{anneal} = 10$, $\mathbf{K} = 2$ and $\tau ol = 0.0000001$) and those determined with the **MCMC** framework. Additionally, the panels contain the synthetic process, used to generate the respective time series, as a reference value. Since the paths $\gamma_i(t)$ resulting from the **MCMC** method are obtained by calculating the mean of a hundred different runs of the Metropolis algorithm, the

deviation from the displayed process is of interest. Therefore, the 100 different single approximations of the optimal process Γ^* are also shown in the graph. We can regard the hidden path determined with the **FEM** as expected value for the infinite number of **MCMC**-realizations. Then the outlines of the deviation shown in the graphs of Figures 5, 6 and 7 can be considered as the confidence intervals of the $\gamma_1(t)$ process calculated by the **MCMC** framework. The big deflection, however, hints at the fact, that the statistics need to be even bigger than 100 to obtain satisfying results.

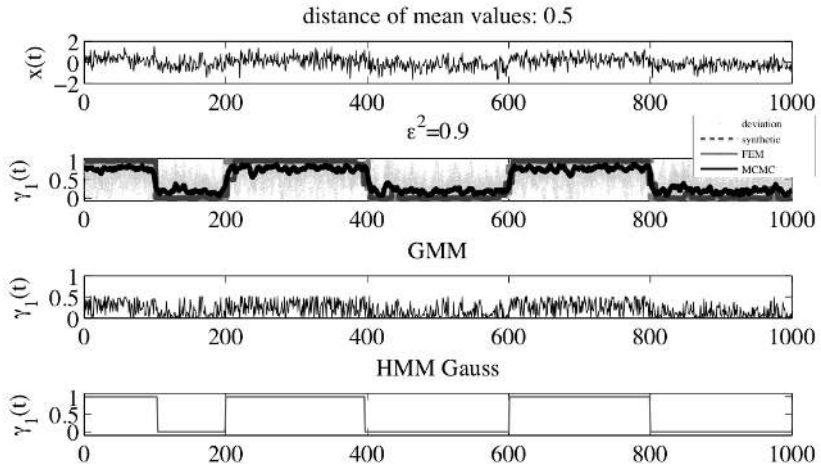

Figure 6: *The one-dimensional time series x_t (upper panel) switches between two normal distributions with expected values $\mu_1 = -0.25$ and $\mu_1 = 0.25$ and equal variance values. In the second panel, the approximations of the optimal hidden process Γ^*, determined with the **MCMC** framework and the **FEM**-methodology, describing the transitions and the persistent states relating to the time series above, are shown. The synthetic path and the 100 different approximations, calculated with the Metropolis algorithm, are also included. The two lower panels display numerical solutions obtained with **GMMs** and **HMM**.*

The graph in the second panel of Figure 6 already illustrates that the accuracy of the results of the **MCMC** technique clearly depends on the data overlap. Clustering quality decrease can be seen even more clearly in the plots of Figure 7. However, it is important to note that despite of the rapid decrease of quality of the results displayed in Figures 6 and 7, the transitions and outlines of the persistent states can still be identified, which should not be taken for granted considering the small distances 0.5 and 0.25 between the clusters and significant cluster variances. In contrast to other cluster analysis methods such as Gaussian mixture Models (**GMMs**) or Hidden Markov Models (**HMM**), the presented **MCMC** method or the **FEM**-clustering framework are able to provide good

approximations of the underlying hidden process when applied to the ill-posed clustering problems like in Figures 6 and 7.

Here **GMMs** are characterized by the underlying assumption that the distribution $f_{x(t)}(s,\Theta)$, relating to the time series $x(t)$ at time t, is a linear combination of $\mathbf{K}$ stationary Gaussian density functions $f_i(s,\theta_i)$:

$$f_{x(t)}(s,\Theta) = \sum_{i=1}^{\mathbf{K}} p_i f_i(s,\theta_i), \tag{32}$$

whereas p_i are weights, which we identify with the probabilities $p_i = P\left[B_t = i|\Theta\right]$ with $B_t = \arg\max_i(\gamma_i(t))$. The expectation-maximization **EM**-algorithm [3] is used to determine the unknown parameters $\theta_1, p_1, \ldots, \theta_{\mathbf{K}}, p_{\mathbf{K}}$. Moreover, it is possible to obtain the conditional probabilities $P\left[B_t = i|\Theta, x_t\right]$. These are generally used to calculate a *Viterbi path* assigning each time t to a cluster parameter θ_i. The path, displayed in the lower panels of Figures 5, 6 and 7, consists of the actual probabilities $P\left[B_t = i|\Theta, x_t\right]$ due to the fact that it is easier to detect the cluster affiliation than regarding the relating Viterbi path. Although it is possible to determine the hidden path with **GMMs** for the least overlapping time series from Figure 5, the algorithm clearly fails to detect transitions between the two clusters in the other two generic ill-posed cases. For example the numerical solution of Figure 7 (third panel from the top) implies that the data of the considered time series is normal distributed meaning that the **EM**-algorithm falsely detects the univariate Gaussianity in this case. A generalization of the **GMMs** in form of Hidden Markov Models (**HMM**) with gaussian observations, however, provides qualitative results for the data with the least overlap and even for the time series with a distance of 0.5 between the mean values μ_i (see lower panel Figure 6 and 5). The cluster affiliation $\gamma_i(t)$ calculated with **HMM** corresponds to the Viterbi path characterizing the model. The set of parameters used here to define the **HMM** also include the transition probabilities $\phi_{ij}(t) = \mathbb{P}(\gamma_j(t)|\gamma_i(t-1))$ which can all be determined according to the normal distributed time series x_t [3]. The **HMM** performance, though good for the small overlap, does not provide a definite assigning process $\gamma_i(t)$ for the data series with a distance of 0.25 of the mean values (see lower panel of Figure 7). Moreover, a major drawback of Bayesian mixture models, here represented by **GMM** and **HMM**, is that the quality of the solution crucially depends on the initialization of the conditional probabilities in the first iteration of the **EM**-algorithm. For example, in context of **HMM** identification via the Expectation-Maximization algorithm (**EM**), setting the initial transition probability matrix to be diagonally dominant may result in the cluster affiliations that are persistent, but it is clear that this type of 'implicit regularization' via

a particular choice of initial parameters is by no means robust since the overall **EM**-algorithm has no control of regularity in it. For the overlapping examples of Figures 5, 6 and 7 we chose the Viterbi path of 50 different runs with the lowest energy value. For both the **GMM** and **HMM** results, the above described 'implicit regularization' (via the setting of initial parameter values) was performed and, as can be seen from Figure 7, both **HMM** and **GMM** methods fail to recover the original persistent path for the most ill-posed scenario with the maximum overlap between the two cluster states. In contrast, both the **MCMC** and the **FEM**-clustering method, due to the build-in explicit $\mathcal{H}^1$-regularization and lack of other implicit probabilistic assumptions, manage to recover the original path with quite satisfactory robustness. Concluding, we may say that it is sensible to apply methods such as the presented **MCMC** method or the **FEM**-clustering instead of the comparatively poorly performing **HMM**/**GMMs** techniques.

Applying a Lilliefors test [29], an adaptation of the Kolmogorov Smirnov test [10], to the time series of Figure 7 also results in the wrong conclusion that the data comes from a single Gaussian distribution, which can lead to a false conjecture that the underlying process does not exhibit a regime-switching behavior. Even for the more separated case of Figure 6 the Lilliefors test does not detect that there are in fact two different normal distributions instead of just one.

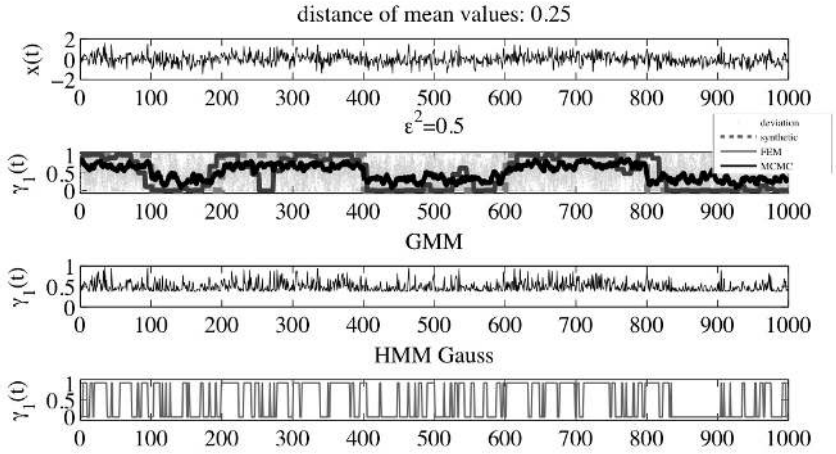

Figure 7: *The figure shows artificially generated data (first panel) and the corresponding resulting paths (second panel) after applying the* ***MCMC*** *framework and the* ***FEM****-technique. The data given in form of a time series is alternating between values of two normal distributions ($\mu_1 = -0.125$ and $\mu_1 = 0.125$, $\sigma_i = 0.5$, $\epsilon^2 = 0.5$). Also the plot displays the synthetic hidden process and* 100 *paths calculated together with the* ***MCMC*** *method with the black colored process being the mean of these results. In the third and fourth panel of the figure the hidden paths determined with* ***GMMs*** *and* ***HMM*** *are shown.*

This point is especially important regarding the fact that these kinds of Gaussianity and regime-behavior tests are commonly applied in computational analysis of physical observation data. As the above results demonstrate, they may fail to recover that there is underlying regime transition behavior in overlapping ill-posed cases. To avoid this problem and to be able to identify the hidden regimes in strongly overlapping data series, it is necessary to make use of analysis methods like the here proposed **MCMC** method or the **FEM**-clustering framework.

4.4 Multidimensional observation data and performance

After we applied the proposed methods to different sets of synthetic time series, we will now investigate its performance on observation data in form of a lattice of daily temperature data [3] from the arctic circle consisting of 100 cells. Firstly, we reduce the space dimension from a hundred to ten with **PCA**. Then we determine the hidden process for different numbers of clusters with the **MCMC** and **FEM**-clustering methods. After that, we find the optimal **K** using the modified Bayesian information criterion defined in (17). Additionally, we apply

[3] Data is provided by `http://data-portal.ecmwf.int/data/d/interim_daily/`

K-Means-clustering [26] to the data to have a comparison with standard techniques. In fact the K-Means-clustering problem formulation is just a special case of the proposed regularized clustering problem (12), i.e., for $\epsilon = 0$, model function (2) and model distance function (5).

The time series has a length of 1095 representing daily measurements from January 2001 till December 2003. We obtain the optimal number of clusters $\mathbf{K} = 3$ (for test runs with $K \in \{1, 2, 3, 4\}$) with the **BIC**. The resulting hidden processes can be seen in Figure 8.

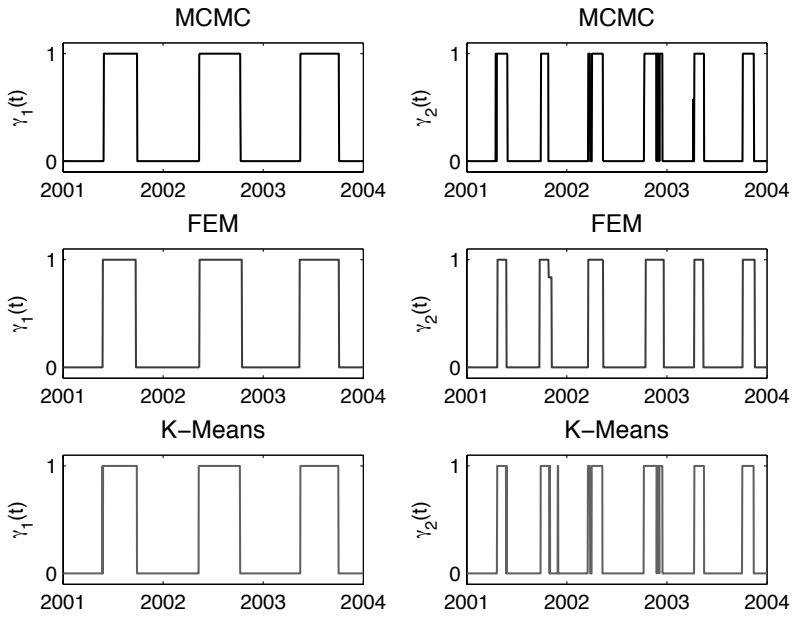

Figure 8: *The figure displays $\gamma_i(t)$ for $i = 1, 2$ obtained with three different techniques (**FEM**, **MCMC**, K-Means). The Metropolis algorithm was set with a Markov chain length of 100000 and $\epsilon = 4$. The **FEM**-technique runs with $N_{FEM-functions} = 100$ and $\epsilon = 4$.*

The transitions between the three clusters can be interpreted as the seasonal changes. Persistency in cluster 1 describes the summer winter cycle and $\gamma_2(t)$ characterizes the transition phases between summer and winter occurring in April and October which can be regarded as the other seasons, fall and spring. All three techniques provide similar approximations of the hidden process only varying in the length of the persistent states. The histograms corresponding to the difference between the observation data and the time series, reproduced with Θ and Γ, display normal distributed behavior in the 2nd to 10th coordinate for the **FEM**-clustering technique and the **MCMC** method. However,

the first coordinate shows uniform distributed histograms for both clustering frameworks. The most important observation is that the difference between the actual data and the predicted time series has a deviation ranging from -50 to 50 in the **MCMC** case and for the **FEM**-clustering method from -110 to 60. This large variance can be explained by looking at the time span, that we identified as *summer*. In these particular intervals the values of the data vary from 55 to 120 which is a wide range to be described with one scalar model parameter component. Nevertheless, the considered clustering frameworks all manage to detect the main seasonal phases of the year and, therefore, provide a good but basic description of the system of interest relating to the given observation data.

We now want to address the performance of the Metropolis algorithm in comparison with the **FEM**-clustering technique applied to higher dimensional data, in particular concerning the run time.

Considering the definition of the dimension of the time series, we distinguish between the *dimension of time*, also referred to as the length of the time series, and the *dimension of space*, referring to the dimension of $x(t)$ at a fixed time t. Obviously, both types of dimension effect the run time and the complexity of the regarded frameworks. We assume that we can reduce the dimension of space with dimension reduction techniques, such as **PCA** [25], and will only concentrate on the effect of the time dimension.

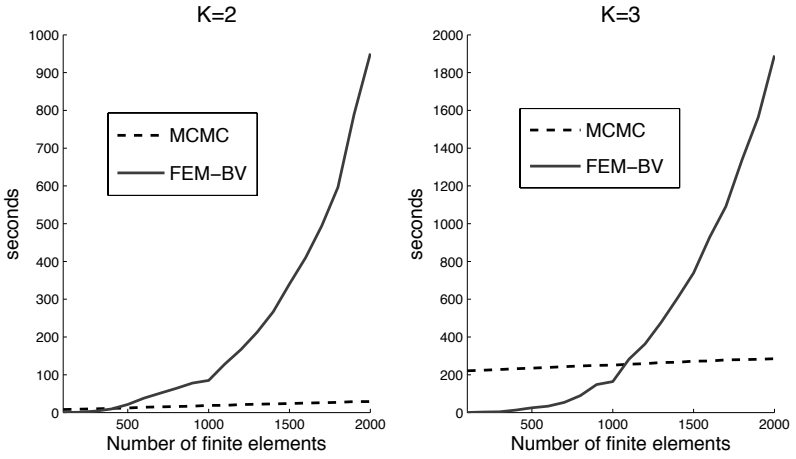

Figure 9: *The panels displays the average computation time in seconds to determine the model parameters γ_i for different numbers of finite elements, i.e., $N_{FEM-functions} = 100, \ldots, 2000$ in steps of 100. The considered synthetic one-dimensional time series have a length of 10000 with 2000 transitions and are generated by sampling from a normal distribution with $\mu_1 = -3.5$ and $\mu_2 = 0$ and variance $\sigma_{1,2} = 0.5$ for $K = 2$ (left panel) and mean values $\mu_1 = -3.5$, $\mu_2 = 0$ and $\mu_3 = 3.5$ again with variance $\sigma_{1,2,3} = 0.5$ (right panel). The **MCMC** method generates 100000 samples and runs with the following parameter settings: initial $\beta = 1$, initial $\nu = 0.1$ and $\epsilon^2 = 0.5$*

Since the performance of the **FEM** method varies significantly for different numbers of finite element functions, it is necessary to consider its run time behavior as a function of $N_{FEM-functions}$. To be able to compare the computation time of the two techniques, we made use of finite elements in the proposed **MCMC**-clustering approach analogue to the **FEM**-clustering method, i.e., sample a discretization $\tilde{\gamma}_i$ of γ_i. Also we will consider the **FEM-BV**-clustering approach due to the fact that our current implementation of the **FEM-BV**-framework [4] is computationally less expensive than our implementation of the **FEM**-clustering method (i.e., with Tikhonov-regularization) proposed in [21].

Since the run time of the **FEM-BV**-clustering is effected by the number of annealing steps and iterations to obtain a good approximation of the global minimum, we will only consider the computational time of one run of the linear optimization problem to find an optimal Γ for a fixed Θ neglecting the usual repetitions (such as annealing steps or iterations for error reduction). The considered synthetic one-dimensional time series have a length of 10000 with 2000 transitions and are generated by sampling from a normal distribution with mean values $\mu_1 = -3.5$ and $\mu_2 = 0$ and variance $\sigma_{1,2} = 0.5$ for $\mathbf{K} = 2$ and mean values $\mu_1 = -3.5$, $\mu_2 = 0$ and $\mu_3 = 3.5$ again with variance $\sigma_{1,2,3} = 0.5$ for $\mathbf{K} = 3$. The finite element **MCMC**-clustering framework is run with 100000 iterations, i.e., the length of the Markov chain is 100000. This number of iterations is sufficiently large to obtain qualitative results even for $N_{FEM-functions} = 2000$. The adaptive simulated annealing is set as follows: initial $\beta = 1$, initial $\nu = 0.1$ and $\epsilon^2 = 0.5$. While for small numbers of finite element functions the **FEM-BV** method still provides solutions in less than a second, the computation time scales badly with a growing number of finite element functions (as can be seen in Figure 9). In contrast to the run time of the **FEM-BV**-clustering technique, the run time of the **MCMC** increases moderately for larger numbers of finite element functions. Even for $N_{FEM-functions} = 10000$, i.e., Γ is sampled without additional discretization, the computational costs are reasonable (115 seconds for $K = 2$ and 542 seconds for $K = 3$). Moreover, it is important to mention that the computationally feasible number of finite elements is limited due to memory capacity problems, i.e., there is a restriction to the level of accuracy that can be provided with the **FEM**-clustering technique.
Note that the synthetic affiliation processes $\Gamma_{\mathbf{K}=2}$ and $\Gamma_{\mathbf{K}=3}$ used to generate the data for Figures 9 each have 2000 regime switches, thus it is not possible to describe the underlying dynamics correctly for $N_{FEM-functions} < 2000$. However, the mean quality level of the results for $N_{FEM-functions} = 2000$ is

[4] The framework with various model distance functions is available on `http://icsweb.inf.unisi.ch/cms/index.php/component/content/article/12-news/77-2012-metstroem-summer-school-qmethods-of-data-analysis-for-fluid-mechanics-meteorology-and-climate-research.html`

sufficient for $\mathbf{K} = 2$:

$$\sum_t \frac{||x(t) - x_{FEM}(t)||}{T} \approx 1.5348 \text{ and } \sum_t \frac{||x(t) - x_{MCMC}(t)||}{T} \approx 1.5433 \quad (33)$$

and for $\mathbf{K} = 3$:

$$\sum_t \frac{||x(t) - x_{FEM}(t)||}{T} \approx 1.5718 \text{ and } \sum_t |\frac{||x(t) - x_{MCMC}(t)||}{T} \approx 1.5860 \quad (34)$$

for both the **FEM-BV**-clustering method (settings: $\tau ol = 0.0000001$ and $\mathbf{C} = 2000$) and the **MCMC**-clustering technique (settings: initial $\beta = 1$, initial $\nu = 0.1$, $\epsilon^2 = 0.5$ and a Markov chain of length 100000). Note that the **FEM-BV**-framework has to be run for more than one (i.e., mean of 13) iteration to obtain qualitative results.

Due to the rapidly growing run time of the **FEM-BV** method it is sensible to apply the **MCMC**-clustering technique in cases where $N_{FEM-functions}$ is considerably big. Usually it is necessary to run the algorithm with more finite element functions as soon as the length of the time series increases but it might also be prudent for data where the persistency of the states is low, i.e., where the transitions between the cluster states are more frequent. Since it is difficult to predict persistence behavior for observational data, it might also be necessary to assume more finite elements to assure results of good quality. Though the **MCMC**-clustering is a good option for applications with longer time series, it is important to mention that with longer time series it might become necessary to increase the number of samples, i.e., the length of the generated Markov chain to gain good approximations. Moreover, the number of considered clusters **K** effects the number of necessary **MCMC** samples. However, in such cases it is possible to further reduce the numerical cost with parallel computing for the Metropolis algorithm [4].

5 Conclusion

A Markov chain Monte Carlo approach to persistent cluster modeling with an adaptive simulated annealing ansatz was presented and its performance was investigated by applying it to different sets of synthetic and observation data and comparing it to standard methods such as **GMM**/**HMM**, K-Means and to the **FEM**-clustering algorithm in terms of efficiency and accuracy. The conceptual advantages of the proposed **MCMC** framework are firstly the good computational performance especially in comparison with the **FEM**-based clustering technique for bigger numbers of finite elements. Moreover, the Metropolis algorithm allows a much higher level of parallelism compared to the **FEM**-based

technique. This opens promising perspectives for parallel high-performance implementations of the method on modern supercomputer architectures. The proposed **MCMC** is also computationally superior when it comes to the feasible number of finite elements which in case of the **FEM**-based clustering is limited due to memory capacity problem. Another import advantage is that the **MCMC** approach does not depend on the choice of the initial parameters, i.e., provides a global optimum of the clustering problem. Also, in contrast with the earlier introduced **FEM**-based clustering methods, the **MCMC** framework allows an uncertainty quantification of the resulting cluster affiliations. Both the **FEM**-based clustering and the **MCMC**-based framework allow to identify a hidden process even for very overlapping data, where standard approaches for regime behavior (like **GMMs** and statistical Gaussianity tests [10, 29]) fail.
The drawback of the current implementation of the **MCMC** method is that it can require careful tuning of three adjustable sampling parameters to obtain a reliable numerical solutions to the problem. In contrast, the **FEM**-clustering technique [20, 21, 22, 24] has only one externally adjustable tolerance parameter. Also, the determination of the optimal length of the underlying Markov chain is a source of uncertainty for the presented **MCMC**-clustering method.
Concluding, the **MCMC** approach provides a good alternative to the existing **FEM**-clustering algorithm for ill-posed clustering problem, i.e. problems with a significant overlap between the clusters. Furthermore the two techniques complement one another regarding run time and quality when applied to a variety of data.

Acknowledgement

The authors thank the DFG SPP 1276 MetStroem "Meteorology and Turbulence Mechanics", Swiss National Research Foundation Grant 200021_131845 "AnaGraph", and the Center for Scientific Simulation (Free University of Berlin, Germany) for funding the research for this paper. We would also like to thank the anonymous referees for many valuable comments.

References

[1] H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19:716 – 723, December 1974.

[2] J. C. Bezdek, R. J. Hathaway, M. J. Sabin, and W. T. Tucker. Convergence theory for fuzzy c-means: Counterexamples and repairs. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(5):873–877, 1987.

[3] J. A. Bilmes. *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*. International computer science institute, Berkeley, CA, 1998.

[4] A. E. Brockwell. Parallel markov chain monte carlo simulation by pre-fetching. *Journal of Computational and Graphical Statistics*, 15:246–261, 2006.

[5] O. Cappé, E. Moulines, and T. Ryden. *Inference in hidden Markov models*. Springer, 2005.

[6] S. Chib and E. Greenberg. Understanding the metropolis-hastings algorithm. *American Statistician*, 49(4):327–335, November 1995.

[7] B. Christiansen. Atmospheric circulation regimes: Can cluster analysis provide the number? *American Meteorological Society*, 20:2229–2250, 2007.

[8] J. de Wiljes, L. Putzig, and I. Horenko. Discrete non-homogenous and non-stationary logistic and markov regression models for spatio-temporal data with unresolved external influences. *submitted to CAMCoS*, 2012.

[9] A. Errahmani, M. Benyakhlef, , and I. Boumhidi. Greenhouse model identification based on fuzzy clustering approach. *ICGST-ACSE Journal*, 9:23–27, 2009.

[10] G. Fasano and A. Franceschini. A multidimensional version of the kolmogorov-smirnov test. *Royal Astronomical Society, Monthly Notices*, 225:155–170, March 1987.

[11] B. Forecasting, D. M. S. S. S. in Statistics Mike West, and J. Harrison. *Bayesian Forecasting and Dynamic Models*. Springer, 1997.

[12] C. Franzke, I. Horenko, A. J. Majda, and R. Klein. Systematic metastable atmospheric regime identification in an agcm. *J. Atmos. Sci.*, 66(9):1997–2016, 2008.

[13] D. Giannakis and A. J. Majda. Quantifying the predictive skill in long range forecasting: Demonstration in a simple ocean model. *Journal of Climate*, submitted Nov 2010.

[14] J. Hadamard. Sur les problèmes aux dérivés partielles et leur signification physique. *Princeton University Bulletin*, 13:49–52, 1902.

[15] L. Hall, A. Bensaid, L. Clarke, R. Velthuizen, M. Silbiger, and J. Bezdek. A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain. *IEEE Transactions on Neural Networks*, 3:672–682, 1992.

[16] L. O. Hall, I. B. Özyurt, and J. C. Bezdek. Clustering with a genetically optimized approach. *IEEE Trans. Evolutionary Computation*, 3(2):103–112, 1999.

[17] W. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

[18] F. Hoeppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. John Wiley and Sons, New York, 1999.

[19] A. Hoerl. Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58:54–59, 1962.

[20] I. Horenko. On robust estimation of low-frequency variability trends in discrete markovian sequences of atmospheric circulation patterns. *Journal of the Atmospheric Sciences*, 66(7):2059–2072, 2009.

[21] I. Horenko. Finite element approach to clustering of multidimensional time series. *SIAM Journal of Scientific Computing*, 32(1):62–83, 2010.

[22] I. Horenko. On clustering of non-stationary meteorological time series. *Dynamics of Atmospheres and Oceans*, 49:164–187, 2010.

[23] I. Horenko. On identification of nonstationary factor models and its application to atmospherical data analysis. *Journal of the Atmospheric Sciences*, 67:1559–1574, 2010.

[24] I. Horenko. Parameter identification in nonstationary markov chains with external impact and its application to computational sociology. *SIAM Mult. Mod. Sym.*, 2010.

[25] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[26] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892, 2002.

[27] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science,*, 220(4598):671–680, 1983.

[28] W. D. Li and C. A. McMahon. A simulated annealing-based optimization approach for integrated process planning and scheduling. *International Journal of Computer Integrated Manufacturing*, 1:80–95, 2007.

[29] H. Lilliefors. On the kolmogorov–smirnov test for normality with mean and variance unknown. *J. Am. Stat. Assoc.*, 62:399–402, 1967.

[30] J. Liu and C. Sabbatti. Simulated sintering: Mcmc with spaces of varying dimensions bayesian statistics. *Oxford University Press*, 6:386–413, 1998.

[31] J. S. Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6:113–119, 1996.

[32] Z. Lui. An image reconstruction algorithm based on tikhonov regularization in electromagnetic tomography. *International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), 2010*, 1:488– 491, 2010.

[33] E. Marinari and G. Parisi. Simulated tempering: a new monte carlo scheme. *Europhys. Lett.*, pages 451–458, 1992.

[34] G. J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, New York, 2000.

[35] Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, June 1953.

[36] P. Metzner, L. Putzig, and I. Horenko. Analysis of persistent non-stationary time series and applications. *CAMCoS*, 2012.

[37] P. D. Moral, A. Jasra, and A. Doucet. Sequential monte carlo samplers. *J. Royal Statist. Soc.*, pages 411–436, 2006.

[38] N. S. Pillai, A. M. Stuart, and A. H. Thiery. Optimal scaling and diffusion iimits for the langevin algorithm in high dimensions. *The Annals of Applied Probability*, 22(6), 2012.

[39] L. Putzig, D. Becherer, and I. Horenko. Optimal allocation of a futures portfolio utilizing numerical market phase-detection. *SIAM J. on Financial Mathematics*, 1:752–779, 2010.

[40] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, New York, 2nd edition, 2004.

[41] G. O. Roberts and J. S. Rosenthal. Optimal scaling for various metropolis–hastings algorithms. *Statistical Science*, 16(4):351–367, 2001.

[42] K. Rose, E. Gurewitz, and G. Fox. A deterministic annealing approach to clustering. *Pattern Recognition Letters*, 11:589–594, 1990.

[43] M. Seel. Atomic clusters and cluster models in solid state physics. *International Journal of Quantum Chemistry: Quantum Chemistry Symposium*, 22(265-274), 1988.

[44] S. Z. Selim and K. Alsulta. A simulated annealing algorithm for the clustering problem. *Pattern Recognition*, 24:1003–1008, 1991.

[45] L.-X. Sun, Y.-L. Xie, X.-H. Song, J.-H. Wang, and R.-Q. Yu. Cluster analysis by simulated annealing. *Computers and Chemistry*, 18(2):103 – 108, 1994.

[46] L. M. Talbot, B. G. Talbot, R. E. Peterson, H.Tolley, and H. D. Mecham. Application of fuzzy grade of membership clustering to analysis of remote sensing data. *Journal of Climate*, 12:200–219, 1999.

[47] A. Tarantola. *Inverse Problem Theory*. Society for Industrial and Applied Mathematics, 2004.

[48] Y. Tian, J. Wu, Z. Wang, and D. Lu. Fuzzy clustering and bayesian information criterion based threshold estimation for robust voice activity detection. *IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings*, 2003.

[49] V. Tola, F. Lillo, M. Gallegati, and R. N. Mantegna. Cluster analysis for portfolio optimization. *Journal of Economic Dynamics and Control*, 32:235–258, 2008.

[50] S. A. Vavasis. Quadratic programming is in np. *Information Processing Letters*, 36(2):73–77, 1990.

[51] G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

[52] P. Wolfe. The simplex method for quadratic programming. *Econometrica*, 27(3):382–398, 1959.

[53] L. Ying, D. Xu, and Z. Liang. On tikhonov regularization for image reconstruction in parallel mri. on tikhonov regularization for image reconstruction in parallel mri. on tikhonov regularization for image reconstruction in parallel mri. *Conf Proc IEEE Eng Med Biol Soc.*, 2004.