

# Chapter 13

## Inferring Leadership from Group Dynamics Using Markov Chain Monte Carlo Methods

**Avishy Y. Carmi, Lyudmila Mihaylova, François Septier, Sze Kim Pang, Pini Gurfil, and Simon J. Godsill**

**Abstract** This chapter presents a novel framework for identifying and tracking dominant agents in groups. The proposed approach relies on a causality detection scheme that is capable of ranking agents with respect to their contribution in recognizing the system's collective behavior based exclusively on the agents' observed trajectories. Further, the reasoning paradigm is made robust to multiple emissions and clutter by employing a class of recently introduced Markov chain Monte Carlo-based group tracking methods. Examples are provided that demonstrate the strong potential of the proposed scheme in identifying actual leaders in swarms of interacting agents and moving crowds.

---

A.Y. Carmi (✉)

Department of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore  
e-mail: [acarmi@ntu.edu.sg](mailto:acarmi@ntu.edu.sg)

L. Mihaylova

School of Computing and Communications, Lancaster University, Lancaster, UK  
e-mail: [mila.mihaylova@lancaster.ac.uk](mailto:mila.mihaylova@lancaster.ac.uk)

F. Septier

Signal Processing and Information Theory Group, TELECOM Lille 1, Villeneuve d'Ascq Cedex, France  
e-mail: [francois.septier@telecom-lille1.eu](mailto:francois.septier@telecom-lille1.eu)

S.K. Pang

DSO National Laboratories, Singapore  
e-mail: [pszekim@dso.org.sg](mailto:pszekim@dso.org.sg)

P. Gurfil

Department of Aerospace Engineering, Technion Israel Institute of Technology, Haifa, Israel  
e-mail: [pgurfil@technion.ac.il](mailto:pgurfil@technion.ac.il)

S.J. Godsill

Department of Engineering, University of Cambridge, Cambridge, UK  
e-mail: [sjg@eng.cam.ac.uk](mailto:sjg@eng.cam.ac.uk)

## 13.1 Introduction

Tracking interacting objects moving in a coordinated fashion and making inference about the patterns of their behavior has been subject of an increased interest in the last decade. Such problems occur in many areas, especially video surveillance, cell tracking in biomedicine, pollutant clouds monitoring and people rescuing. The common pattern of the whole group is of main interest, not the individual trajectories on their own. In most of the multi-object tracking methods, as opposed to groups tracking methods, tracking of individual objects is the common approach. This is an especially challenging problem when the groups are composed of hundreds or thousands elements and the inference needs to be done quickly, in real time, based on heterogeneous multi-sensor data.

*Groups* can be considered as structured objects, a term which reflects the interrelationships between their components. These endogenous forces give rise to group hierarchies and are instrumental in producing emergent phenomena. Fortunately, these are exactly the factors essential for maintaining coordination within and between groups, a premise which to some extent allows us to treat them as united entities in a high level tracking paradigm. Any knowledge of existence of such interrelations facilitates sophisticated agent-based behavioral modeling which, in practice, comprises of a set of local interaction rules or mutually interacting processes (e.g., Boids system [31], causality models [17, 30]) – an approach which by itself provides insightful justifications of characteristic behaviors in the fundamental subsystem level and likewise of group hierarchies and emergent social patterns (see [30]).

### 13.1.1 Reasoning About Behavioral Traits

Being the underlying driving mechanism for evoking emergent phenomena, hierarchies and principal behavior patterns, the ingrained interactions between agents are possibly the most pivotal factors that should be scrutinized in high level scene understanding. Such interrelations can take the form of a causal chain in which an agent's decisions and behavior are affected by its neighbors and likewise have either direct or indirect influence on other agents. The ability to fully represent these interrelations based exclusively on passive observations such as velocity and position, lays the ground for the development of sophisticated reasoning schemes that can potentially be used in applications such as activity detection, intentionality prediction, and artificial awareness.

In this work we demonstrate this concept by developing a causality reasoning framework for ranking agents with respect to their cumulative contribution in shaping the collective behavior of the system. In particular, our framework is able to distinguish leaders and followers based exclusively on their observed trajectories.

### 13.1.2 *Novelties and Contributions*

The contribution of this work is twofold. Firstly, a novel *causality* reasoning scheme is derived for ranking agents with respect to their decision-making capabilities (dominance) as substantiated by the observed emergent behavior. Dominant agents in that sense are considered to have a prominent influence on the collective behavior and are experimentally shown to coincide with actual leaders in groups. Secondly, the causality scheme is consolidated with a recently introduced Markov chain Monte Carlo (MCMC)-based particle method [9, 28] for tracking agents and group hierarchies in potentially cluttered environments.

The subsequent Sects. 13.1.3–13.2 provide an overview of existing group tracking schemes with an emphasis on the underlying MCMC-based particle methods.

The remaining part of this chapter is organized in the following way. Section 13.3 develops the causality-driven agent ranking approach. Section 13.4 demonstrates the performance of the causality identification scheme using a few illustrative examples. Finally, concluding remarks and some open issues are discussed in Sect. 13.5.

### 13.1.3 *Multiple Group Tracking*

Over the past decade various methods have been developed for group tracking. These can be divided into two broad classes, depending on the underlying complexities: (1) methods for a relatively small number of groups, with a small number of group components [15, 24, 28], and (2) methods for groups comprised of hundreds or thousands of objects (normally referred to as cluster/crowd tracking techniques) [2, 9]. In the second case the whole group is usually considered as an extended object (an ellipse or a circle) which center position is estimated, together with the parameters of the extent.

Different models of groups of objects have been proposed in the literature, such as particle models for flocks of birds [19], and leader-follower models [26]. However, estimating the dynamic evolution of the group structure has not been widely studied in the literature, although there are similarities with methods used in evolving network models [1, 11].

Typically tracking many objects (hundreds or thousands) can be solved by clustering techniques or other methods where the aggregated motion is estimated, as it is in the case of vehicular traffic flow prediction/estimation, with fluid dynamics type of models combined with particle filtering techniques [27]. For thousands of objects forming a group, the only possible solution is to consider them as an extended object. The extended object tracking problem reduces then to joint state and parameter estimation.

Estimation of parameters in general nonlinear non-Gaussian state-space models is a long-standing problem. Since particle filters (PFs) are known with the challenges they face for parameter estimation and for joint state and parameter estimation [4],

most solutions in the literature split the problems into two parts: (i) state estimation, followed by (ii) parameter estimation (see e.g., [3]). In [3] an extended object tracking problem is solved when the static parameters are estimated using Monte Carlo methods (data augmentation and particle filtering), whereas the states are estimated with a Mixture Kalman filter or with an interacting multiple model filter.

### 13.1.3.1 PFs for Tracking in Variable State Dimensions

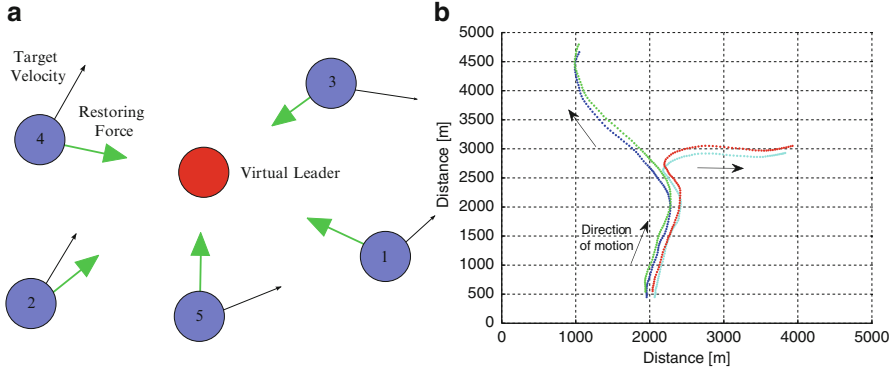
An extension of the PF technique to a varying number of objects is introduced in [28, 34] and [24]. In [34] a PF implementation of the probability hypothesis density (PHD) filter is derived. This algorithm maintains a representation of the filtering belief mass function using random set realizations (i.e., particles of varying dimensions). The samples are propagated and updated based on a Bayesian recursion consisting of set integrals. Both works of [28] and [24] develop a MCMC PF scheme for tracking varying numbers of interacting objects. The MCMC approach outperforms the conventional PF due to its efficient sampling mechanism. Nevertheless, in its traditional non-sequential form it is inadequate for sequential estimation. The techniques used by Pang et al. [28] and Khan et al. [24] amend the MCMC for sequential filtering (see also [5]). The work in [24] copes with inconsistencies in state dimension by utilizing the reversible jump MCMC method introduced in [18]. In [28], on the other hand the computation of the marginal filtering distribution is avoided as in [5]. The algorithm operates on a fixed dimension state space through indicator variables for labeling of active object states (the two approaches are essentially equivalent).

## 13.2 Models and Algorithms for Group Tracking

This section briefly reviews the fundamental concepts underlying the MCMC-based group tracking approaches in [28] and [9].

### 13.2.1 *Virtual Leader Model*

The idea of group modeling is to adopt a behavioral model in which each member of a group interacts with the other members of the group, typically making its velocity and position more similar to that of others in the same group. In [28], this idea has been conveniently formulated in continuous time through a multivariate stochastic differential equation (SDE) and then derived in discrete time without approximation errors, owing to the assumed linear and Gaussian form of the model.



**Fig. 13.1** Group model with virtual leader – illustration of the restoring forces (a) and of a single realization showing a group of four objects that splits into two groups of two objects (b)

In particular, two different models have been proposed. In the first, the basic group model, the group parameter is modeled as a deterministic function of the objects. In the second, the group model with a virtual leader, an additional state variable is introduced in order to model the bulk or group parameter. This second approach is closer in spirit to the bulk velocity model and virtual leader-follower model [26]. Such model provides a more flexible behavior since the virtual leader is no longer a deterministic function of the individual object states. Figure 13.1 gives a graphical illustration of the restoring forces towards the virtual leader for a group of five objects.

The spatio-temporal structure for the  $i$ th object in a group, as defined in [28], is given by:

$$d\mu_{t,i}^x = \{-\alpha[\mu_{t,i}^x - v_t^x] - \gamma_1 \dot{\mu}_{t,i}^x - \beta[\dot{\mu}_{t,i}^x - \dot{v}_t^x] + r_i\} dt + \sigma_x dW_{t,i}^x \quad (13.1)$$

$$d\dot{v}_t^x = -\gamma_2 \dot{v}_t^x dt + \sigma_g dG_t^x \quad (13.2)$$

Here  $\mu_{t,i}^x$  is the Cartesian position in the  $X$  direction of the  $i^{\text{th}}$  object in the group at time  $t$ , with  $\dot{\mu}_{t,i}^x$  the corresponding velocity.  $v_t^x$  and  $\dot{v}_t^x$  represent respectively the Cartesian position and the velocity both in the  $X$  direction of the unobserved virtual leader of the group.  $W_{t,i}^x$  and  $G_t^x$  are two independent standard Brownian motions.  $W_{t,i}^x$  is assumed to be independently generated for each object  $i$  in the group, whereas  $G_t^x$  is a noise component common to all members of a group. The parameters  $\alpha$  and  $\beta$  are positive, and reflect the strength of the pull towards the group bulk. The “mean reversion” terms  $\gamma_1 \dot{\mu}_{t,i}^x$  and  $\gamma_2 \dot{v}_t^x$  simply prevent the velocities of the object and the virtual leader drifting up to very large values with time. Finally, in order to reduce or eliminate behavior in which objects become colocated or collide spatially, which are clearly infeasible or highly unlikely in practice, an additional repulsive force  $r_i$  is introduced in (13.1) when objects become too close.

### 13.2.2 Modeling Groups of Extended Objects

In practice, objects may produce more than a single emission, and in some cases they may indeed consist of many individual entities moving in a coordinated fashion (i.e., clusters). Such scenarios normally involve additional extent parameters that embody the potentially dynamic physical boundary of an object. In this respect, the fairly simple idea adopted in [9] represents a dynamically evolving group of extended objects, which are otherwise referred to as clusters, by means of a time-varying Gaussian mixture model (i.e., each mixture component corresponds to an individual object). In what follows, we briefly review the essentials of this approach.

Assume that at time  $k$  there are  $l_k$  clusters, or targets at unknown locations. Each cluster may produce more than one observation yielding the measurement set realization  $\mathbf{z}_k = \{\mathbf{y}_k(i)\}_{i=1}^{m_k}$ , where typically  $m_k \gg l_k$ . At this point we assume that the observation concentrations (clusters) can be adequately represented by a parametric statistical model.

Letting  $\mathbf{z}_{0:k} = \{\mathbf{z}_0, \dots, \mathbf{z}_k\}$  be the measurement history up to time  $t_k$ , the cluster tracking problem may be defined as follows. We are concerned with estimating the posterior distribution of the random set of unknown parameters, i.e.  $p(x_k | \mathbf{z}_{0:k})$ , from which point estimates for  $x_k$  and posterior confidence intervals can be extracted.

For reasons of convenience we consider an equivalent formulation of the posterior that is based on existence variables. Thus, following the approach adopted in [9] the random set  $x_k$  is replaced by a fixed dimension vector coupled to a set of indicator variables  $e_k = \{e_k^j\}_{j=1}^n$  showing the activity status of elements (i.e.,  $e_k^j = 1$ ,  $j \in [1, n]$  indicates the existence of the  $j$ th element where  $n$  stands for the total number of elements). To avoid possible confusion, in what follows we maintain the same notation for the descriptive parameter set  $\mathbf{x}_k$  which is now of fixed dimension.

In [9], each cluster is modeled via a Gaussian pdf. Following this only the first two moments, namely the mean and covariance, need to be specified for each cluster (under these restrictions, the cluster tracking problem is equivalent to that of tracking an evolving Gaussian mixture model with a variable number of components). It is worth mentioning, that the approach itself does not rely on the Gaussian assumption and other parameterized density functions could equally be adopted in this framework. Thus,

$$\mathbf{x}_k^j = \{\mu_k^j, \dot{\mu}_k^j, \Sigma_k^j, w_k^j, \rho_k^j\}, \quad \mathbf{x}_k = \{\mathbf{x}_k^j\}_{j=1}^n, \quad (13.3)$$

where  $\mu_k^j$ ,  $\dot{\mu}_k^j$ ,  $\Sigma_k^j$  and  $w_k^j$  denote the  $j$ th cluster's mean, velocity, covariance and associated unnormalized mixture weight at time  $k$ , respectively. The additional parameter  $\rho_k^j$  denotes the local turning radius of the  $j$ th cluster's mean at time  $k$ .

### 13.2.3 Sequential Inference Using MCMC-Based PF

The group tracking problems discussed above can be efficiently solved via the MCMC-based particle method initially proposed for solution of group tracking problems in [28]. This method aims at sequentially approximating the following joint posterior distribution

$$p(\mathbf{x}_k, \mathbf{x}_{k-1} | \mathbf{z}_{0:k}) \propto p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{0:k-1}) \quad (13.4)$$

where the state vector  $\mathbf{x}_k$  comprises of the objects' instantaneous position, velocity and extent parameters at time  $t_k$ . In what follows we would refer to the (discrete) time  $t_k$  as simply  $k$ .

Since the closed form expression of the distribution  $p(\mathbf{x}_{k-1} | \mathbf{z}_{0:k-1})$  is generally unknown, the proposed scheme approximates it by using a set of unweighted particles

$$p(\mathbf{x}_{k-1} | \mathbf{z}_{0:k-1}) \approx \frac{1}{N} \sum_{j=1}^N \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{(j)}) \quad (13.5)$$

where  $N$  is the number of particles,  $\delta(\cdot)$  is the Dirac delta, and  $(j)$  is the particle index. Then, by plugging this particle approximation into (13.4), an appropriate MCMC scheme can be used to draw from the joint posterior distribution  $p(\mathbf{x}_k, \mathbf{x}_{k-1} | \mathbf{z}_{0:k})$ . The converged MCMC outputs are then extracted to give an empirical approximation of the posterior distribution of interest at time  $k$ , thus seeding the next step of the filtering at time  $k+1$ .

At the  $m$ th MCMC iteration, the following procedure is performed to obtain samples from  $p(\mathbf{x}_k, \mathbf{x}_{k-1} | \mathbf{z}_{0:k})$ :

1. Make a joint draw for  $\{\mathbf{x}_k, \mathbf{x}_{k-1}\}$  using a Metropolis Hastings step,
2. Update successively some elements in  $\mathbf{x}_k$  by using a series of Metropolis Hastings-within-Gibbs.

#### 13.2.3.1 Metropolis Hastings Step for the Cluster Tracking Problem

The Metropolis Hastings (MH) algorithm generates samples from an aperiodic and irreducible Markov chain with a predetermined (possibly unnormalized) stationary distribution. This is a constructive method which specifies the Markov transition kernel by means of acceptance probabilities based on the preceding time outcome. As part of this, a proposal density is used for drawing new samples. In our case, setting the stationary density as the joint filtering pdf of the object states  $\mathbf{x}_k, \mathbf{x}_{k-1}$  and the corresponding indicator variables  $e_k, e_{k-1}$ , i.e.,  $p(\mathbf{x}_k, e_k, \mathbf{x}_{k-1}, e_{k-1} | \mathbf{z}_{0:k})$  (of which the marginal is the desired filtering pdf), a new set of samples from this distribution can be obtained after the MH burn-in period. This procedure is described next.

---

**Algorithm 3** MCMC particle filtering algorithm
 

---

1. Given previous time samples  $\{\mathbf{x}_{k-1}^{(i)}, e_{k-1}^{(i)}\}_{i=1}^N$  perform the following steps
2. for  $i = 1, \dots, N + N_{\text{Burn-in}}$
3. for  $j = 1, \dots, n$
4. Cluster evolution: Simulate

$$(\hat{\mu}_{k-1}^{j(i)}, \hat{\mu}_{k-1}^{j(i)}, \hat{\Sigma}_{k-1}^{j(i)}, w_{k-1}^{j(i)}, \rho_{k-1}^{j(i)}) \longrightarrow (\hat{\mu}_k^{j(i)}, \hat{\mu}_k^{j(i)}, \hat{\Sigma}_k^{j(i)}, w_k^{j(i)}, \rho_k^{j(i)})$$

5. end for
  6. Perform MCMC move (Algorithm 4).
  7. Draw a new set of indicators  $e_k^{j(i)}$ ,  $j = 1, \dots, n$  for the accepted move.
  8. Perform Gibbs refinement (Algorithm 5).
  9. end for
- 

First, we simulate a sample from the joint propagated pdf  $p(\mathbf{x}_k, e_k, \mathbf{x}_{k-1}, e_{k-1} \mid \mathbf{z}_{0:k-1})$  by drawing

$$(\mathbf{x}'_k, e'_k) \sim p(\mathbf{x}_k, e_k \mid \mathbf{x}'_{k-1}, e'_{k-1}) \quad (13.6)$$

where  $(\mathbf{x}'_{k-1}, e'_{k-1})$  is uniformly drawn from the empirical approximation

$$\hat{p}(\mathbf{x}_{k-1}, e_{k-1} \mid \mathbf{z}_{0:k-1}) = N^{-1} \sum_{i=1}^N \delta(\mathbf{x}_{k-1}^{(i)} - \mathbf{x}_{k-1}) \delta(e_{k-1}^{(i)} - e_{k-1}) \quad (13.7)$$

This sample is then accepted or rejected using the following Metropolis rule.

Let  $(\mathbf{x}_k^{(i)}, e_k^{(i)}, \mathbf{x}_{k-1}^{(i)}, e_{k-1}^{(i)})$  be a sample from the realized chain of which the stationary distribution is the joint filtering pdf. Then the MH algorithm accepts the new candidate  $(\mathbf{x}'_k, e'_k, \mathbf{x}'_{k-1}, e'_{k-1})$  as the next realization from the chain with probability

$$\gamma = \min \left\{ 1, \frac{p(\mathbf{z}_k \mid \mathbf{x}'_k, e'_k, m_k)}{p(\mathbf{z}_k \mid \mathbf{x}_k^{(i)}, e_k^{(i)}, m_k)} \right\} \quad (13.8)$$

where  $p(\mathbf{z}_k \mid \mathbf{x}'_k, e'_k, m_k)$  is the likelihood function. The converged output of this scheme simulates the joint density  $p(\mathbf{x}_k, e_k, \mathbf{x}_{k-1}, e_{k-1} \mid \mathbf{z}_{0:k})$  of which the marginal is the desired filtering pdf.

It has already been noted that the above sampling scheme may be inefficient in exploring the sample space as the underlying proposal density of a well behaved system (i.e., of which the process noise is of low intensity) introduces relatively small moves. This drawback is alleviated here by using a secondary Gibbs refinement stage [9].

A single cycle of the basic MCMC cluster tracking algorithm of [9] is summarized in Algorithms 3, 4, and 5.



**Algorithm 4** MCMC move

- 
1. Compute the MH acceptance probability  $\gamma$  of the new move using (13.8).
  2. Draw  $u \sim U[0, 1]$
  3. if  $u < \gamma$
  4. Accept  $\mathbf{s}^{(i)} = (\mathbf{x}_k^{(i)}, e_k^{(i)}, \mathbf{x}_{k-1}^{(i)}, e_{k-1}^{(i)})$  as the next sample of the realized chain.
  5. else
  6. Retain  $\mathbf{s}^{(i)} = \mathbf{s}^{(i-1)}$ .
  7. end if
- 

**Algorithm 5** Particles refinement (Metropolis within Gibbs)

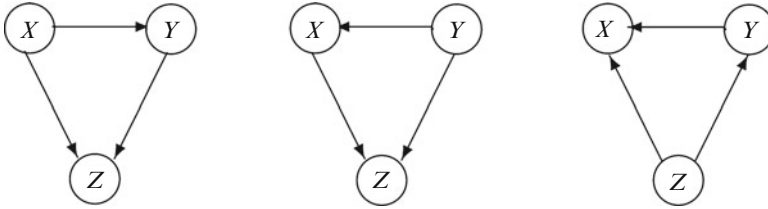
- 
1. for  $j = 1, \dots, n$
  2. if  $e_k^{j,(i)} = 1$
  3. for  $l = 1, \dots, N_{\text{MH Steps}}$
  4. Propose a move  $\bar{\mu}_k^j$ .
  5. Compute the MH acceptance probability  $\bar{\gamma}$  of the new move.
  6. Draw  $u \sim U[0, 1]$
  7. if  $u < \bar{\gamma}$
  8. Accept the new move by setting  $\mu_k^{j,(i)} = \bar{\mu}_k^j$ .
  9. else
  10. Retain previous  $\mu_k^{j,(i)}$ .
  11. end if
  12. end for
  13. end if
  14. end for
- 

**13.2.3.2 Multiple Chain and Evolutionary MCMC**

The theory of multiple chain MCMC grasps that a mixing mechanism for synthesizing samples across chain realizations is necessary for improving robustness to the well known practical problem of quasi-ergodicity otherwise known as poor mixing. Existing multiple chain approaches, such as parallel tempering [13, 14], evolving population particle filters [6–8, 21, 22, 29] and population MCMC [23, 25], utilize exchange mechanisms to expedite convergence. The evolutionary MCMC approach, on the other hand, incorporates an additional structure for generating possibly improved candidates based on convergent chain realizations. This method has been proved successful in high dimensional settings. An evolutionary extension of the basic MCMC filtering scheme is provided in the Appendix part of this work.

**13.3 Causality-Driven Agent Ranking**

The so-called probabilistic approach to causality, which has reached maturity over the past two decades (see for example Pearl [30], Geffner [12], and Shoam [32] for an extensive overview), establishes a convenient framework for reasoning and inference of causal relations in complex structural models.



**Fig. 13.2** From *left to right*: depiction of the causal hierarchies (based on out degrees)  $(X, Y, Z)$ ,  $(Y, X, Z)$ , and  $(Z, Y, X)$ . The most influential agents in the causal diagrams from *left to right* are  $X$ ,  $Y$  and  $Z$ , respectively

Many notions in probabilistic causality rely extensively on structural models and in particular on causal Bayesian networks which are normally referred to as simply causal networks (CN's). A CN is a directed acyclic graph compatible with a probability distribution that admits a Markovian factorization and certain structural restrictions [30].

### 13.3.1 Causal Hierarchies

In this work the term *causal hierarchies* refers to ranking of agents with respect to their cumulative effect on the actions of the remaining constituents in the system. The word “causal” here reflects the fact that our measure of distinction embodies the intensity of the causal relations between the agent under inspection and its counterparts. Adopting the information-theoretic standpoint, in which the links of a CN are regarded as information channels [10], one could readily deduce that the total effect of an agent is directly related to the local information flow entailed by its corresponding in and out degrees. To be more precise, the total effect of an agent is computed by summing up the associated path coefficients (obtained by any standard Bayesian network learning approach) of either inward or outward links. This concept is further illustrated in Fig. 13.2.

### 13.3.2 Inferring Causal Hierarchies via PCA

To some extent, causal hierarchies can be inferred using the class of principal component analysis (PCA)-based methods. Probably the most promising one in the context of our problem is the multi-channel singular spectrum analysis (M-SSA), which is otherwise known as extended empirical orthogonal function (EEOF) analysis [16]. The novel approach we suggest has some relations with M-SSA. The relevant details, however, are beyond the scope of this work. A performance evaluation of both our method and M-SSA is provided in the numerical study part in the following sections.

### 13.3.3 Structural Dynamic Modeling Approach

Structural equation modeling is commonly used for representing the underlying links of a CN [30]. In our case, this formulation assumes a rather dynamic form (i.e., comprising of multiple time series of the agents' observed traits such as velocity and position)

$$\mathbf{x}_k^i = \sum_{j \neq i} \sum_{m=1}^p \alpha^{j \rightarrow i}(m) \mathbf{x}_{k-m}^j + \varepsilon_k^i, \quad i = 1, \dots, n \quad (13.9)$$

where  $\{\mathbf{x}_k^i\}_{k=0}^{\infty}$  and  $\{\varepsilon_k^i\}_{k=0}^{\infty}$  denote the  $i$ th random process and a corresponding white noise driving sequence, respectively. The coefficients  $\{\alpha^{j \rightarrow i}(m)\}_{m=1}^p$  quantify the causal influence of the  $j$ th process on the  $i$ th process. Notice that the Markovian model (13.9) has a finite-time horizon of the order  $p$  (also referred to as the wake parameter). In the standard multivariate formulation, the coefficients  $\alpha^{j \rightarrow i}(m)$  are square matrices of an appropriate dimension. For maintaining a reasonable level of coherency we assume that these coefficients are scalars irrespectively of the dimension of  $\mathbf{x}_k^i$ . Nevertheless, our arguments throughout this Section can be readily extended to the standard multivariate case.

The methodology underlying the so-called Granger causality [17] considers an F-test of the null hypothesis  $\alpha^{j \rightarrow i}(m) = 0$ ,  $m = 1, \dots, p$  for determining whether the  $j$ th process G-causes the  $i$ th process. The key idea here follows the simple intuitive wisdom that the more significant these coefficients are, the more likely they are to reflect a causal influence. In the framework of CNs the causal coefficients are related to the conditional dependencies within the probabilistic network, which in turn implies that their values can be learned based on the realizations of the time series  $\{\mathbf{x}_k^i\}_{k=0}^{\infty}$ ,  $i = 1, \dots, n$ . In what follows, we demonstrate how the knowledge of these coefficients allows us to infer the fundamental role of individual agents within the system. Before proceeding, however, we shall define the following key quantity.

**Definition 13.1 (Causation Matrix).** The causal influence of the process  $\mathbf{x}^j$  on the process  $\mathbf{x}^i$  can be quantified by

$$A_{ij} = \sum_m [\alpha^{j \rightarrow i}(m)]^2 \geq 0. \quad (13.10)$$

In the above definition,  $A_{ij}$  denotes the coefficient relating the two processes  $\mathbf{x}^j$  and  $\mathbf{x}^i$  so as to suggest an overall matrix structure that would provide a comprehensive picture of the causal influences among the underlying processes. The matrix  $A = [A_{ij}] \in \mathbb{R}^{n \times n}$ , termed the *causation matrix*, essentially quantifies the intensity of all possible causal influences within the system (note that according to the definition of a CN, the diagonal entries in  $A$  vanish). It can be easily recognized that a single row in this matrix exclusively represents the causal interactions

affecting each individual process. Similarly, a specific column in  $A$  is comprised of the causal influences of a single corresponding process on the entire system. This premise motivates us to introduce the notion of total causal influence.

**Definition 13.2 (Total Causal Influence Measure).** The total causal influence (TCI)  $T_j$  of the process  $\mathbf{x}_k^j$  is obtained as the  $l_1$ -norm of the  $j$ th column in the causation matrix  $A$ , that is

$$T_j = \sum_{i=1}^n |A_{ij}| = \sum_{i=1}^n A_{ij} \tag{13.11}$$

Having formulated the above concepts we are now ready to elucidate the primary contributions of this work, both of which rely on the TCI measure defined above.

### 13.3.4 Dominance and Similarity

A rather intuitive, but nonetheless striking, observation about the TCI is that it essentially reflects the dominance of each individual process in producing the underlying emergent behavior. This allows us to decompose any complex act into its prominent behavioral building blocks (processes) using a hierarchical ordering of the form

$$\text{Least dominant } T_{j_1} \leq T_{j_2} \leq \dots \leq T_{j_n} \text{ Most dominant} \tag{13.12}$$

Equation (13.12) is given an interesting interpretation in the application part of this work, where the underlying processes  $\{\mathbf{x}_k^j\}_{j=1}^n$  correspond to the motion of individual agents within a group. In the context of this example, the dominance of an agent is directly related to its leadership capabilities. By using the TCI measure it is therefore possible to distinguish between leaders and followers.

Another interesting implication of the TCI is exemplified in the following argument. Consider the two extreme processes in (13.12), one of which is the most dominant,  $\mathbf{x}_k^{j_n}$ , while the other is the least dominant,  $\mathbf{x}_k^{j_1}$ . Now, suppose we are given a new process  $\mathbf{x}_k^i$ ,  $i \neq j_1, j_n$  and are asked to assess its dominance based exclusively on the two extremals, with respect to the entire system. Then, a common intuition would suggest to categorize  $\mathbf{x}_k^i$  as a dominant process in the system whenever it resembles  $\mathbf{x}_k^{j_n}$  more than  $\mathbf{x}_k^{j_1}$  in the sense of  $|T_{j_n} - T_i| < |T_{j_1} - T_i|$  and vice versa. This idea is summarized below.

**Definition 13.3 (Causal Similarity).** A process  $\mathbf{x}_k^j$  is said to resemble  $\mathbf{x}_k^i$  more than  $\mathbf{x}_k^l$  if and only if  $|T_j - T_i| < |T_j - T_l|$ .

In the context of the previously-mentioned example, we expect that dominant agents with high leadership capabilities would possess similar TCIs that would distinguish them from the remaining agents, the followers.

### 13.3.5 Bayesian MCMC Estimation of $\alpha^{j \rightarrow i}$

In typical applications the coefficients  $\alpha^{j \rightarrow i}(m)$ ,  $m = 1, \dots, p$  in (13.9) may be unknown. Providing that the realizations of the underlying processes are available it is fairly simple to estimate these coefficients by treating them as regressors. Such an approach by no means guarantees an adequate recovery of the underlying causal structure (see the discussion about the identifiability of path coefficients and a related assertion concerning non-parametric functional modeling in [30] pp. 156–157, both have a clear connotation to the “fundamental problem of causal inference” [20]). Nevertheless, it provides a computationally efficient framework for making inference in systems with exceptionally large number of components. This premise is evident by noting from (13.9) that while fixing  $i$  the coefficients  $\alpha^{j \rightarrow i}(m)$ ,  $\forall j \neq i$ ,  $m = 1, \dots, p$  are statistically independent of  $\alpha^{j \rightarrow l}(m)$ ,  $\forall l \neq i$ .

In a Bayesian framework we confine the latent causal structure by imposing a prior on the coefficients  $\alpha^{j \rightarrow i}(m)$ . Let  $p_\alpha^i$  and  $p_\alpha^{j \rightarrow i}$  be the priors of  $\{\alpha^{j \rightarrow i}(m), \forall j \neq i\}$ , and  $\alpha^{j \rightarrow i}(m)$ , respectively. Let also  $p_\varepsilon^i$  be some prescribed (not necessarily Gaussian) probability density of the white noise in (13.9). Then,

$$\begin{aligned} p(\{\alpha^{j \rightarrow i}(m), \forall j \neq i\} | \mathbf{x}_{0:k}^{1:n}) &\propto \\ p_\alpha^i \prod_{t=p}^k p(\mathbf{x}_t^i | \{\alpha^{j \rightarrow i}(m), \mathbf{x}_{t-p:t-1}^j, \forall j \neq i\}) & \\ = p_\alpha^i \prod_{t=p}^k p_\varepsilon^i(\mathbf{x}_t^i - \sum_{j \neq i, m=1}^p \alpha^{j \rightarrow i}(m) \mathbf{x}_{t-m}^j), \quad i = 1, \dots, n & \quad (13.13) \end{aligned}$$

where  $\mathbf{x}_{0:k}^{1:n} = \{\mathbf{x}_0^1, \dots, \mathbf{x}_0^n, \dots, \mathbf{x}_k^1, \dots, \mathbf{x}_k^n\}$ , and  $\mathbf{x}_{t-p:t-1}^j = \{\mathbf{x}_{t-p}^j, \dots, \mathbf{x}_{t-1}^j\}$ . A viable estimation scheme for  $\alpha^{j \rightarrow i}(m)$  which works well in most generalized settings is a Metropolis-within-Gibbs sampler that operates either sequentially or concurrently on the conditionals

$$\begin{aligned} p(\alpha^{j \rightarrow i}(m) | \mathbf{x}_{0:k}^{1:n}, \{\alpha^{l \rightarrow i}, \forall l \neq j, i\}) &\propto \\ p_\alpha^{j \rightarrow i} \prod_{t=p}^k p(\mathbf{x}_t^i | \{\alpha^{l \rightarrow i}(m), \mathbf{x}_{t-p:t-1}^l, \forall l \neq i\}) & \quad (13.14) \end{aligned}$$

The obtained estimates at time  $k$  are then taken as the average of the converged chain (i.e., subsequent to the end of some prescribed burn-in period).

### 13.3.6 Causal Reasoning in Cluttered Environments

In many practical applications the constituent underlying traits, which are represented here by the processes  $\{\mathbf{x}_k^j\}_{j=1}^n$ , may not be perfectly known (in the context

of our work these could be the object position and velocity,  $\mu_k^j, \dot{\mu}_k^j$ ). Hence instead of the actual traits one would be forced to use approximations that might not be consistent estimates of the original quantities (e.g.,  $\hat{\mu}_k^j, \hat{\dot{\mu}}_k^j$ ). As a consequence, the previously suggested structure might cease being an adequate representation of the latent causal mechanism. A plausible approach for alleviating this problem is to introduce a compensated causal structure that takes into account the exogenous disturbances induced by the possibly inconsistent estimates. Such a model can be readily formulated as a modified version of (13.9), that is

$$\hat{\mu}_k^i = \sum_{j \neq i, m=1}^p \alpha^{j \rightarrow i}(m) \hat{\mu}_{k-m}^j + \varepsilon_k^i + \zeta_k^i, \quad i = 1, \dots, n, \quad (13.15)$$

where the additional factor  $\zeta_k^i$  denotes an exogenous bias. Hence, one can use (13.15) to predict the effects of interventions in  $\zeta_k^i$  directly from passive observations (which are taken as an output of a tracking algorithm, e.g.,  $\hat{\mu}_k^j$  or  $\hat{\dot{\mu}}_k^j$ ) without adjusting for confounding factors. See [30] (p. 166) for further elaborations on the subject.

### 13.4 Illustrative Examples

We demonstrate the performance of our suggested reasoning methodology and some of the previously mentioned concepts using both synthetic and realistic examples. All the scenarios considered here involve a group of dynamic agents, some of which are leaders that behave independently of all others. The leaders themselves may exhibit a highly nonlinear and non-predictive motion pattern which in turn affects the group's emergent behavior. We use a standard CN (13.9) with a predetermined time horizon  $p$  for disambiguating leaders from followers based exclusively on their instantaneous TCIs. In all cases the processes  $\mathbf{x}_k^i$ ,  $i = 1, \dots, n$  are taken as either the increment  $\dot{\mu}_k^i$  or position  $\mu_k^i$  of each individual agent in the group. In addition, the unified tracking and reasoning paradigm is demonstrated by replacing the actual position and increment with the corresponding outputs of the MCMC cluster tracking algorithm,  $\hat{\mu}_k^i$  and  $\hat{\dot{\mu}}_k^i$ .

The performance of the causality inference scheme is directly related to its ability to classify leaders based on their TCI values. As leaders are, by definition, more dominant than followers in some measure space, essentially shaping the overall group behavior, we expect that their TCI values would reflect this fact. Furthermore, the hierarchy (13.12) should allow us to disambiguate them from the remaining agents according to the notion of causal similarity which was introduced in Sect. 13.3.4. Following this argument we define a rather distinctive performance measure which allows us to assess the aforementioned qualities.

Let  $G$  be a set containing the leaders indices, i.e.,

$$G = \{j \mid x_k^j \text{ is a leader's instantaneous position or velocity}\}.$$

Let also  $v$  be a vector containing the agents' ordered indices according to the instantaneous hierarchy at time  $k$

$$T_{j_1} \leq \dots \leq T_{j_n}, \quad (13.16)$$

i.e.,  $v = [j_n, \dots, j_1]^T$ . Having stated this we can now define the following *performance index*

$$e = \max\{i \in [1, n] \mid v_i \in G\} \quad (13.17)$$

The above quantity indicates the worst TCI ranking of a leader. As an example, consider a case with, say, five leaders. Then the best performance index we could expect would be five, implying that all leaders have been identified and were properly ranked according to their TCIs. If the performance index yields a value greater than 5, say 10, it implies that all leaders are ranked among the top 10 agents according to their TCIs. The performance index cannot go below the total number of leaders and cannot exceed the total number of agents.

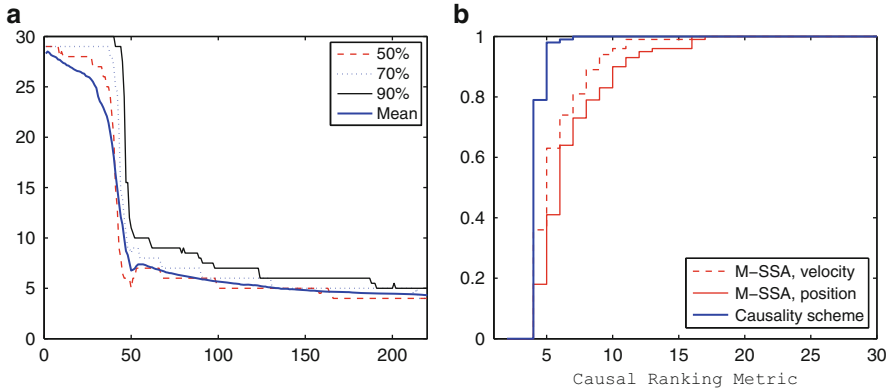
### 13.4.1 *Swarming of Multiple Interacting Agents (Boids)*

Our first example pertains to identification of leaders and followers in a dynamical system of multiple interacting agents, collectively performing in a manner usually referred to as *swarming* or *flocking*.

In the current example, Reynolds-inspired flocking [31] is used to create a complex motion pattern of multiple agents. Among these agents, there are leaders, who independently determine their own position and velocity, and followers, who interact among themselves and follow the leader agents.

The inference scheme performance over 100 Monte Carlo runs, in which the agents initial state and velocity were randomly picked, is provided in Fig. 13.3. The synthetic scenario considered consists of 30 agents, 4 of which are actual leaders. The performance index cumulative distribution function (CDF) for this scenario, which is illustrated via the 50, 70 and 90 percentile lines, is shown over the entire time interval in the left panel in this figure. The percentiles indicate how many runs out of 100 yielded a performance index below a certain value. Thus, 50% of the runs yielded a performance index below the 50 percentile, 70% of the runs attained values below the 70 percentile, and so on. Following this, it can be readily recognized that from around  $k = 150$  the inference scheme is able to accurately identify the actual leaders in 50% of the runs. A further examination of this figure reveals that the 4 actual leaders are ranked among the top 6 from around  $k = 180$  in 90% of the runs.

A comparison of leaders ranking capabilities of the proposed approach with that of the M-SSA method is provided in the right panel in Fig. 13.3. The instantaneous CDFs of both techniques are shown when using either position or



**Fig. 13.3** Identification performance over time (abscissa) of the causality scheme (*left*) and the ranking CDF at time  $t = 220$  (*right*) of both the causality scheme and the M-SSA method (using either velocity or position data) based on 100 Monte Carlo runs. **(a)** Causal ranking. **(b)** Causal ranking CDF

velocity time series data. This figure clearly demonstrates the superiority of the proposed approach with respect to the M-SSA.

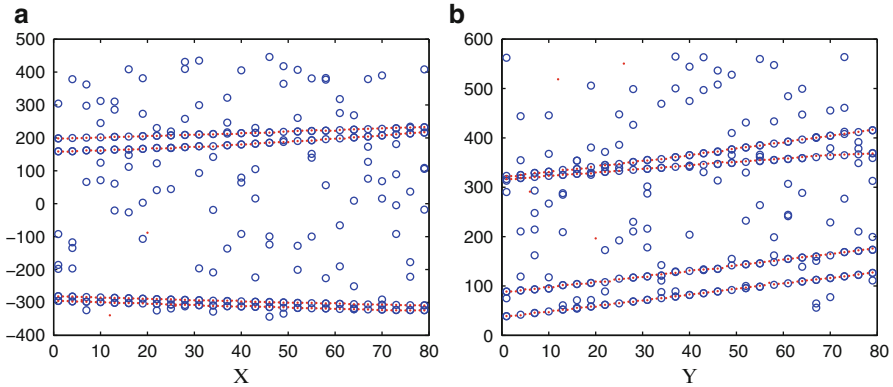
### 13.4.2 Identifying Extended Leaders in Clutter

In the following example the actual agent tracks are replaced by the output of an MCMC-based tracking approach that was initially derived in [9, 28] and is briefly described in Sect. 13.2. The scenario consists of four agents out of which two are leaders. As before we use the Boids system for simulating the entire system. This time, however, the produced trajectories are contaminated with clutter and additional points representing multiple emissions from possibly the same agent (i.e., agents are assumed to be extended objects). These observations are then used by the MCMC tracking algorithm of which the output is fed to the causality detection scheme, in a fashion similar to the one described in Sect. 13.3.6.

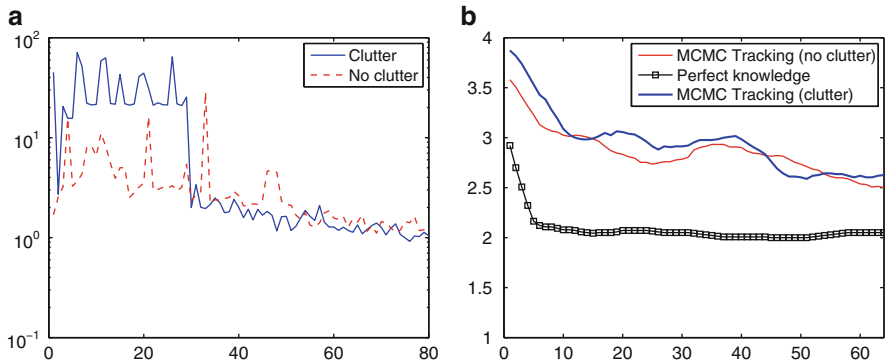
The tracking performance of the MCMC algorithm is demonstrated both in Fig. 13.4 and in the left panel in Fig. 13.5. In Fig. 13.4, the estimated tracks and the cluttered observations are shown for a typical run. The averaged tracking performance of the MCMC approach is further illustrated based on 20 Monte Carlo runs using the Hausdorff distance [9] in Fig. 13.5. From this Figure it can be seen that the mean tracking errors become smaller than 1 after approximately 50 time steps in either cases of cluttered and non-cluttered observations.

The averaged leaders ranking performance in this example is illustrated for three different scenarios in the right panel in Fig. 13.5. Hence, it can be readily recognized





**Fig. 13.4** Point observations and estimated tracks over time (abscissa). **(a)** X. **(b)** Y

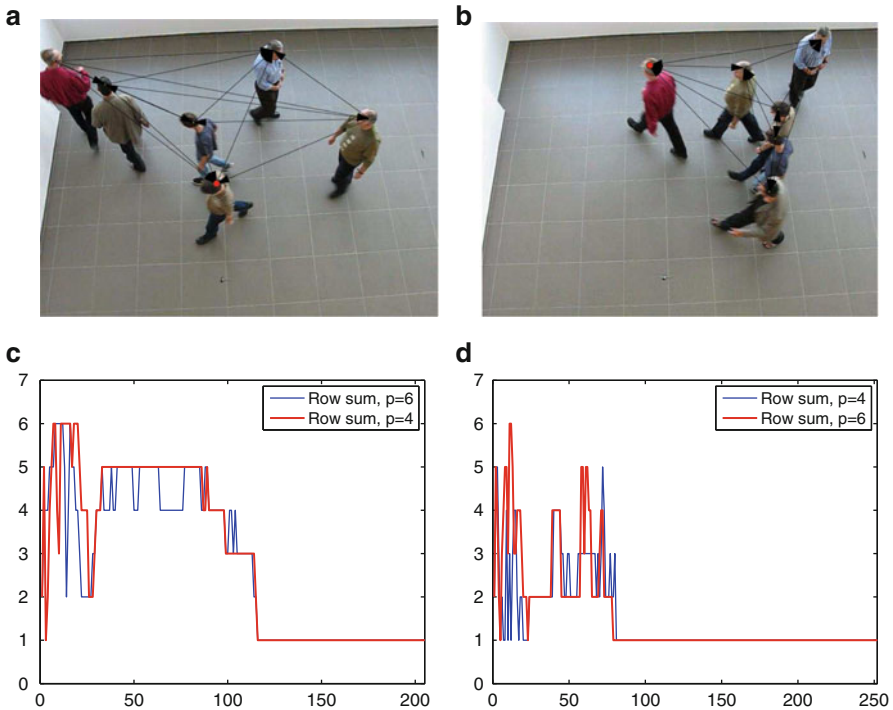


**Fig. 13.5** Tracking performance and causality ranking over time (abscissa) averaged over 20 Monte Carlo runs. **(a)** Hausdorff distance. **(b)** Causality ranking

that the two leaders are accurately identified after approximately 10 time steps when the agent positions are perfectly known. As expected, this performance is deteriorated in the presence of clutter and multiple emissions, essentially attaining an averaged ranking metric of nearly 2.5 after 60 time steps.

### 13.4.3 Identifying Group Leaders from Video Data

Our third, more practical example, deals with the following application. Consider a group of people, among which there are subgroups of leaders and followers. The followers coordinate their paths and motion with the leader. Using video observations only of the group, determine who the group leaders are. To that end,



**Fig. 13.6** Reconstructed instantaneous causal diagrams shown with the corresponding video frames (*upper panel*), and causality ranking performance over time (*lower panel*). (a) Video 1. (b) Video 2. (c) Video 1. (d) Video 2

one must first develop a procedure for estimating the trajectories of  $n$  people from a given video sequence. The input to the described procedure is a movie with  $n$  moving people, where  $n$  is known. The objective is to track each person along the frame sequence, and then feed this information into the CN mechanism for inferring the leaders and followers.

As we are dealing with a rather noiseless and non-cluttered scenario, a simple k-means clustering was used to recover individual person tracks from SIFT (scale-invariant feature transform) features. This approach was applied to two different video sequences in which there were five followers and one leader. Snapshots are shown in the upper panel in Fig. 13.6. In these videos, the actual leader (designated by a red shirt) performs a random trajectory, and the followers loosely follow its motion pattern. The clustering procedure described above is used to estimate the trajectories of the objects (the trajectories were filtered using a simple moving-average procedure to reduce the amount of noise contributed by the k-means clustering method). These trajectories were fed into the causality inference scheme.

The results of this procedure are shown in the bottom panel in Fig. 13.6, which depicts the causality performance index for two values of the finite-time horizon

(wake parameter),  $p$ . It is clearly seen that from a certain time point the algorithm identifies the *actual* leader in both videos irrespective to the value of  $p$ .

### 13.5 Concluding Remarks

A novel causal reasoning framework has been proposed for ranking agents with respect to their contribution in shaping the collective behavior of the system. The proposed scheme copes with clutter and multiple emissions from extended agents by employing a Markov chain Monte Carlo group tracking method. This approach has been successfully applied for identifying leaders in groups in both synthetic and realistic scenarios.

## Appendix

### *Evolutionary MCMC Implementation*

The basic MH scheme can be used to produce several chain realizations each starting from a different (random) state. In that case, the entire population of the converged MH outputs (i.e., subsequent to the burn-in period) approximates the stationary distribution. Using a population of chains enjoys several benefits compared to a single-chain scheme. The multiple-chain approach can dramatically improve the diversity of the produced samples as different chains explore various regions that may not be reached in a reasonable time when using a single chain realization [23, 25]. Furthermore, having a population of chains facilitates the implementation of interaction operators that manipulate information from different realizations for improving the next generation of samples.

Following the approach of [33], the evolutionary MCMC cluster tracking algorithm uses genetic operators to generate new samples. The decoding scheme used here simply transforms the samples into their binary representations.

Let  $\mathcal{G}_l = \{\mathbf{x}_k^{(i)}, e_k^{(i)}, \mathbf{x}_{k-1}^{(i)}, e_{k-1}^{(i)}\}_{i=1}^N$  be the  $l$ th realization of the converged chain at time  $k$ . Define by

$$\mathcal{G} := \{\mathcal{G}_1, \dots, \mathcal{G}_L\} \quad (13.18)$$

the entire population set consisting of  $L$  chain realizations. In order to produce an improved generation of  $N$  samples from the joint filtering pdf, members of the population  $\mathcal{G}$  undergo two successive genetic operations: crossover and mutation.

## Chromosomes and Sub-chromosomes

Any genetic manipulation act on a unique data structure known as a chromosome which usually takes the form of a string. Here, a chromosome refers to a binary representation of a particle  $(\mathbf{x}_k^{(i)}, e_k^{(i)})$ . Since every particle consists of several clusters endowed with their own individual properties,  $\mu_k^{j,(i)}$ ,  $\dot{\mu}_k^{j,(i)}$ ,  $\Sigma_k^{j,(i)}$ ,  $w_k^{j,(i)}$  and  $\rho_k^{j,(i)}$ , in practice a chromosome consists of several concatenated binary strings each corresponding to a distinct property of a certain cluster. In this work, we term sub-chromosomes, the strings pertaining to individual properties. Assuming there are no more than  $n$  clusters for which there are exactly 5 properties yields a chromosome that is built up of  $5n$  sub-chromosomes. The active sub-chromosomes within a chromosome are those that belong to active clusters, i.e., clusters for which  $e_k^{j,(i)} = 1$ ,  $j = 1, \dots, n$ .

## The Crossover Operator

The crossover works by switching genetic material between two parent samples taken from two different chain realizations for producing an offspring. The two parents,  $(\mathbf{x}_k, e_k)_1$  and  $(\mathbf{x}_k, e_k)_2$  are independently drawn from  $\hat{p}(\mathbf{x}_k, e_k | z_{0:k})$ , i.e., they are picked uniformly at random from the population  $\mathcal{G}$ . The sub-chromosomes  $A$  and  $B$  corresponding to the same property in the chosen parents are then manipulated as follows. For every  $r \in [1, r_s]$ , where  $r_s$  denotes the string length of either  $A$  or  $B$ , the bits  $A_r$  and  $B_r$  are swapped with some predetermined probability  $\beta$ . The resulting offspring sub-chromosomes are then encoded to produce two new candidates  $(\mathbf{x}'_k, e'_k)_1$  and  $(\mathbf{x}'_k, e'_k)_2$ . At this point an additional MH step is performed for deciding whether the new offspring will be a part of the improved population. This step is crucial for maintaining an adequate approximation of the target distribution. In order to ensure that the resulting chain is reversible, on acceptance both new candidates should replace their parents, otherwise both parents should be retained [33].

Following the above argument, it can be easily verified that the acceptance probability of both offspring is [33]

$$\min \left\{ 1, \left( \frac{1-\beta}{\beta} \right)^a \frac{\hat{p}((\mathbf{x}'_k, e'_k)_1 | z_{0:k}) \hat{p}((\mathbf{x}'_k, e'_k)_2 | z_{0:k})}{\hat{p}((\mathbf{x}_k, e_k)_1 | z_{0:k}) \hat{p}((\mathbf{x}_k, e_k)_2 | z_{0:k})} \right\} \quad (13.19)$$

where  $a$  denotes the total number of swapped bits.

## The Mutation Operator

The mutation operator flips the  $r$ th bit within a given chromosome with probability  $\beta_m$ . Let  $(\mathbf{x}_k, e_k)$  be a sample drawn from  $\hat{p}(\mathbf{x}_k, e_k | z_{0:k})$  (i.e., picked uniformly

**Algorithm 6** Evolutionary MCMC cluster tracking

1. Execute Algorithm 3 for  $L$  chains in parallel.
2. Subsequent to the end of the burn in period proceed as follows for every chain sample.
3. *Interaction*: Perform genetic operations to obtain an improved offspring
  - a. Picking uniformly at random two distinct chain realizations perform crossover between their latest accepted samples.
  - b. Compute the acceptance probability of the two offspring using (13.19).
  - c. Accept the new offspring accordingly or retain both parent samples.
  - d. Mutate the accepted samples of either chains and compute the associated acceptance probability (13.20).
  - e. Accept the mutated sample accordingly.

at random from the population  $\mathcal{G}$ . Then, it can be verified that the acceptance probability of a mutated candidate  $(\mathbf{x}'_k, e'_k)$  is [33]

$$\min \left\{ 1, \left( \frac{1 - \beta_m}{\beta_m} \right)^a \frac{\hat{p}(\mathbf{x}'_k, e'_k | z_{0:k})}{\hat{p}(\mathbf{x}, e_k | z_{0:k})} \right\} \quad (13.20)$$

where  $a$  denotes the total number of bits changed.

A single cycle of the evolutionary MCMC filtering scheme is summarized in Algorithm 6.

## References

1. Albert, R., Barabási, A.-L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**(1), 47–97 (2002)
2. Ali, S., Shah, M.: Floor fields for tracking in high density crowd scenes. In: *Computer Vision – ECCV 2008*. Volume 5303 of *Lecture Notes in Computer Science*, pp. 1–14. Springer, Berlin/Heidelberg (2008)
3. Angelova, D., Mihaylova, L.: Extended object tracking using Monte Carlo methods. *IEEE Trans. Signal Process.* **56**(2), 825–832 (2008)
4. Arulampalam, M., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **50**(2), 174–188 (2002)
5. Berzuini, C., Nicola, G., Gilks, W.R., Larizza, C.: Dynamic conditional independence models and Markov chain Monte Carlo methods. *J. Am. Stat. Assoc.* **92**(440), 1403–1412 (1997)
6. Bhaskar, H., Mihaylova, L.: Combined data association and evolving population particle filter for tracking of multiple articulated targets. *EURASIP J. Image Video Process.* **2011**, article ID 642532 (2011)
7. Bhaskar, H., Mihaylova, L., Maskell, S.: Population-based particle filters. In: *Proceedings of the from the Institution of Engineering and Technology (IET) Seminar on Target Tracking and Data Fusion: Algorithms and Applications*, Birmingham, pp. 31–38 (2008)
8. Cappé, O., Guillin, A., Marin, J.-M., Robert, C.P., Roberty, C.P.: Population Monte Carlo. *J. Comput. Gr. Stat.* **13**, 907–929 (2004)
9. Carmi, A., Septier, F., Godsill, S.J.: The Gaussian mixture MCMC particle algorithm for dynamic cluster tracking. In: *Proceedings of the 12th International Conference on Information Fusion*, pp. 1179–1186. Seattle, WA (2009)

10. Cheng, J., Greiner, R., Kelly, J., Bell, D., Liu, W.: Learning Bayesian network from data: an information-theory based approach. *Artif. Intell.* **137**(1–2), 43–90 (2002)
11. Doroogovtsev, S.N., Mendes, J.F.F.: Evolution of networks. *Adv. Phys.* **51**, 1079–1187 (2002)
12. Geffner, H.: Default Reasoning: Causal and Conditional Theories. MIT, Cambridge (1992)
13. Geyer, C.: Markov chain maximum likelihood. In: Keramigas, E. (ed.) *Computing Science and Statistics: The 23rd Symposium on the Interface*. Interface Foundation, Fairfax (1991)
14. Geyer, C., Thompson, E.A.: Annealing Markov chain Monte Carlo with applications to ancestral inference. *J. Am. Stat. Assoc.* **90**, 909–920 (1995)
15. Gning, A., Mihaylova, L., Maskell, S., Pang, S.K., Godsill, S.: Group object structure and state estimation with evolving networks and Monte Carlo methods. *IEEE Trans. Signal Process.* **12**(2), 523–536 (2011)
16. Golyandina, N., Nekrutkin, V., Zhigljavsky, A. (eds.): *Analysis of Time Series Structure: SSA and Related Techniques*. Chapman and Hall, Boca Raton (2001)
17. Granger, C.W.J.: Investigating causal relations by econometric models and cross-spectral methods. *Econometrica* **37**, 424–438 (1969)
18. Green, P.J.: Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* **82**(4), 711–732 (1995)
19. Helbing, D.: Traffic and related self-driven many-particle systems. *Rev. Mod. Phys.* **73**, 1067–1141 (2002)
20. Holland, P.W.: Statistics and causal inference. *J. Am. Stat. Assoc.* **81**, 945–960 (1986)
21. Iba, Y.: Population-based Monte Carlo algorithms. *J. Comput. Gr. Stat.* **13**(4), 175–193 (2000)
22. Iba, Y.: Population Monte Carlo algorithms. *Trans. Jpn. Soc. Artif. Intell.* **16**, 279 (2000)
23. Jasra, A., Stehphens, D.A., Holmes, C.C.: Population-based reversible jump Markov chain Monte Carlo. *Biometrika* **94**(4), 787–807 (2007)
24. Khan, Z., Balch, T., Dellaert, F.: MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(11), 1805–1819 (2005)
25. Liu, J.S.: *Monte Carlo Strategies in Scientific Computing*. Springer, New York (2001)
26. Mahler, R.: *Statistical Multisource-Multitarget Information Fusion*. Artech House, Boston (2007)
27. Mihaylova, L., Boel, R., Hegyi, A.: Freeway traffic estimation within recursive Bayesian framework. *Automatica* **43**(2), 290–300 (2007)
28. Pang, S.K., Li, J., Godsill, S.J.: Detection and tracking of coordinated groups. *IEEE Trans. Aerosp. Electron. Syst.* **47**(1), 472–502 (2011)
29. Pantrigo, J., Sánchez, A., Gianikellis, K., Monteymayor, A.S.: Combining particle filter and population based metaheuristics for visual articulated object tracking. *Electron. Lett. Comput. Vis. Image Anal.* **5**(3), 68–83 (2005)
30. Pearl, J.: *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK (2000)
31. Reynolds, C.W.: Flocks, herds, and schools: a distributed behavioral model. *Comput. Gr.* **21**, 25–34 (1987)
32. Shoam, Y.: *Reasoning About Change: Time and Causation from the Standpoint of Artificial Intelligence*. MIT, Cambridge (1988)
33. Strens, M.: Evolutionary MCMC sampling and optimization in discrete spaces. In: *Proceedings of the Twentieth International Conference on Machine Learning*, Washington, DC (2003)
34. Vo, B., Singh, S., Doucet, A.: Sequential Monte Carlo methods for multi-target filtering with random finite sets. *IEEE Trans. Aerosp. Electron. Syst.* **41**(4), 1224–1245 (2005)