Robust Online Modeling of Counts in Agent Networks

Radomír Žemlička and Kamil Dedecius

***Abstract*—Many real-world processes of interest produce nonnegative integer values standing for counts. For instance, we count packets in computer networks, people in monitored areas, or particles incident on detectors. Often, the ultimate goal is the modeling of these counts. However, standard techniques are computationally demanding and sensitive to the amount of available information. In our quest to solve the objective, we consider two prominent features of the contemporary world: online processing of streaming data, and the rapidly evolving ad-hoc agent networks. We propose a novel algorithm for a collaborative online estimation of the zero-inflated Poisson mixture models in diffusion networks. Its main features are low memory and computational requirements, and the capability of running in inhomogeneous networks. There, the agents possibly observe different processes, and locally decide which of their neighbors provide useful information. Two simulation examples demonstrate that the algorithm attains good stability and estimation performance even under slowly varying parameters.**

***Index Terms*—Collaborative estimation, diffusion, distributed estimation, excessive zeros, Poisson regression.**

I. Introduction

FULLY distributed *online* modeling of various stochastic processes from *streaming* data in networks with information diffusion is an established discipline in the signal processing domain [1], [2], [3]. Its applications can be found, e.g., in the Internet of Things, social networks, sensor grids, and a variety of other networked scenarios [4], [5], [6], [7], [8], [9]. We focus on fully distributed modeling algorithms. They assume that the agents degree is generally higher than two and that the communication is limited to the adjacent neighbors. Typically, two collaboration strategies are distinguished. First, the consensus strategy with the goal of reaching a (mostly global) agreement about the variables of interest. This necessarily relies on intermediate iterations among agents between two subsequent time instants. Second, the diffusion strategy, where any type of information is exchanged at most once per each time instant. Both strategies have their advantages and disadvantages, and numerous modifications have been proposed in the last decade. Many details can be found in the books [1], [2], and [9]. They are an excellent source of relevant information and references about the topic.

In the signal processing domain, the main focus has naturally been given to the omnipresent continuous variables. This gave rise to the diffusion recursive least squares [10], Bernoulli filters [11], least mean squares [12], [13], [14], [15], particle filters [16], [17], [18], or Kalman filters [19], [20]. Many variants, modifications, and improvements stem from these basic algorithms. The popularity of these filters ultimately implicates their somewhat controversial use for discontinuous variables. Indeed, they *may* perform relatively well despite violating some fundamental assumptions, such as the noise distribution and the response variable support. For instance, the continuous data models are routinely used for discrete counts, i.e., data that take nonnegative integer values. The results (predictions, estimates) are acceptable as long as the observed values are high enough. However, such models are doomed to fail if the counts tend to be low or if there are excessive zeros [21], [22].

In this paper, we will specifically focus on count models. They are notably important in epidemiology, finance, transportation, physics, or networking [21], [23]. They mostly belong to the class of the generalized linear models (GLMs) that link the explanatory variables (regressors) with the observations through convenient link functions [24], [25]. The count models employ the logarithmic link function that gives rise to the Poisson regression model. Unfortunately, the nonlinear link function is not without costs – the vast majority of GLMs (including the Poisson GLM) cannot be inferred without demanding numerical or Markov chain Monte Carlo (MCMC) methods [25], [26]. This is a critical handicap in modern applications processing fast streaming data, for instance, in online counting of crowds, packets, or particles. The first Poisson regression algorithm suitable for low-cost sequential (online) modeling of counts was proposed recently by the authors [27].

A frequent issue of the Poisson GLM is its limited use in many engineering, medical, or sociology applications violating its basic assumptions. Empirical count data typically exhibit *overdispersion*, i.e., the variance of the observed variable exceeds the mean. Its typical source is the existence of multiple sub-populations (sub-processes) generating the observed data [28]. Second, the real-world count data are often subject to *zero-inflation*. They contain excessive zeros that cannot be explained by the Poisson model [29], [30]. The typical sources

Manuscript received 2 July 2022; revised 5 December 2022 and 24 March 2023; accepted 1 April 2023. Date of publication 6 April 2023; date of current version 19 April 2023. This work was supported by the Grant Agency of the Czech Technical University in Prague, under Grant SGS20/213/OHK3/3T/18. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Usman A Khan. *(Corresponding author: Kamil Dedecius.)*

Radomír Žemlička is with the Faculty of Information Technology, Czech Technical University in Prague, 160 00 Prague, Czech Republic (e-mail: zemlirad@fit.cvut.cz).

Kamil Dedecius is with the Institute of Information Theory and Automation, Czech Academy of Sciences, 182 00 Prague, Czech Republic (e-mail: dedecius@utia.cas.cz).

Digital Object Identifier 10.1109/TSIPN.2023.3264990

of excessive zeros are missing observations, sensor/target coverings, dropouts, dead times etc. Both the overdispersion and zero-inflation completely rule out the generic Poisson GLMs. A way around both problems consists in mixture-based modeling. The overdispersion calls for mixtures of Poisson GLMs, while the zero-inflation introduces a non-Poissonian component accounting for the excessive zeros [28], [31], [32], [33]. Similarly to the basic Poisson GLM, the inference of mixture models heavily relies on demanding *offline* numerical methods. Furthermore, complex statistical models, e.g., the mixture models, share one key characteristic: the need for a relatively large amount of observations for their inference. This often leads to deployment of multiple (possibly varied) sensors in practice. Ultimately, they form an agent (sensor) network [1]. Real-world examples comprise crowd counting using several cameras with different fields of view, spatially distributed traffic counting with optical, pneumatic, and inductive devices, or particle counting with different measuring principles [34], [35], [36]. The increasingly popular distributed information processing, however, may then face significant challenges due to the network *inhomogeneity*. For instance, sensors with wide fields of view may observe processes that are invisible to sensors with narrow fields of view. Similarly, cameras operating in the infrared spectrum may observe phenomena that are not observable in the visible spectrum. As a result, the set of observed and modeled processes (e.g., mixture components) of one agent may partially or even completely differ from the corresponding sets of other agents. Reliable identification of common information is inevitable.

In this paper, we carefully focus on the discussed issues, namely the overdispersion, zero-inflation, and network-based information processing. The considerable novelty consists of several aspects. We propose a novel algorithm for a numerically stable online inference of the zero-inflated Poisson mixture model (ZIPMM) from streaming data. It is formulated as a mixture estimation task, where the excessive zeros are ascribed to a Dirac distribution located at the origin. Other sub-processes are modeled by Poisson components. Finally, the idea of improving the modeling performance by collaboration of multiple agents is brought to life for networks with information diffusion. The resulting algorithm admits the inhomogeneity of the network, where the agents possibly observe partially or completely different processes. The automated identification of common mixture components is a part of the solution. This allows for the deployment of the proposed method to real-world applications, where the state-of-the-art solutions rely on demanding centralized information processing.

The paper is structured as follows: The problem is formulated in Section II. In Section III, the algorithm for the collaborative inference is developed. The information fusion over the network is described in the subsequent Section IV. Section V discusses some properties and limitations of the algorithm. Section VI demonstrates the algorithm performance on two simulation examples. Finally, Section VII concludes the paper.

II. Problem Formulation

Let us assume a network consisting of a set of agents $\mathcal{I} = \{1, \ldots, I\}$. Its structure can be represented by a connected graph where the agents correspond to vertices and the undirected edges to the communication channels among them. The communication is spatially limited to one edge distance. That is, for each agent $i \in \mathcal{I}$ we identify its closed neighborhood $\mathcal{I}^i \subset \mathcal{I}$ of agents whose information is at disposal to i. For convenience, the agent i belongs to $\mathcal{I}^i$ too.

The agents $i \in \mathcal{I}$ are interested in a *sequential* (online) discrete-time modeling of an observable streaming stochastic process $\{Y_t^i; t = 1, 2, \ldots\}$ of nonnegative integer counts. We denote their own mutually independent observations by y_t^i. If there exist explanatory variables, a convenient model for Y_t^i is the zero-inflated Poisson mixture model (ZIPMM). It consists of K^i Poisson components (GLMs), and one Dirac component. The Poisson components link the observations y_t^i with related known regressors $x_t^i \in \mathbb{R}^n$, and corresponding unknown vectors $\beta_{k,t}^i \in \mathbb{R}^n, k = 1, \ldots, K^i$ of regression coefficients. The Dirac component is located at zero and accounts for excessive zero measurements that remain unexplained by the Poisson components. Typically, these zeros correspond to deadlocks, missing measurements, failures, dead times, dropouts, sensor/target covering etc.

At each agent $i \in \mathcal{I}$, the components are assigned unknown nonnegative relative weights $\phi_{0,t}^i, \ldots, \phi_{K^i,t}^i$ taking values in the unit K^i simplex. In particular, $\phi_{0,t}^i$ corresponds to the probability that the variable Y_t^i is produced by the Dirac component, and $\phi_{1,t}^i, \ldots, \phi_{K^i,t}^i$ correspond to the probabilities that it is produced by one of the K^i Poisson components. To summarize, the local ZIPMM at the agent i has the general form

$$Y_t^i | x_t^i, \boldsymbol{\beta}_t^i, \boldsymbol{\phi}_t^i \sim \phi_{0,t}^i Dir(0) + \sum_{k=1}^{K^i} \phi_{k,t}^i Pois\Big(\underbrace{\exp\left(\beta_{k,t}^{i,\intercal} x_t^i\right)}_{\text{rate parameter}} \Big), \tag{1}$$

where the unknown parameters are summarized by $\boldsymbol{\phi}_t^i = [\phi_{0,t}^i, \ldots, \phi_{K^i,t}^i]^\intercal$ and $\boldsymbol{\beta}_t^i = \{\beta_{1,t}, \ldots, \beta_{K^i,t}\}$. Our ultimate goal is their reliable estimation. The probability density function of the model (1) reads

$$f^i(y_t^i | x_t^i, \boldsymbol{\beta}_t^i, \boldsymbol{\phi}_t^i) = \phi_{0,t}^i \delta_0 + \sum_{k=1}^{K^i} \phi_{k,t}^i f_k^i(y_t^i | x_t^i, \beta_{k,t}^i), \tag{2}$$

where δ_0 is the Dirac delta distribution at 0, and $f_k^i(\cdot)$ are the Poisson densities with the rate parameter plugged in,

$$f_k^i(y_t^i | x_t^i, \beta_{k,t}^i) = \frac{\exp(\beta_{k,t}^{i,\intercal} x_t^i y_t^i - \exp(\beta_{k,t}^{i,\intercal} x_t^i))}{y_t^i!}. \tag{3}$$

By linearity of the expectation operator and equality of the Poisson rate parameter to the first moment it follows that

$$\mathbb{E}[Y_t^i | x_t^i, \boldsymbol{\beta}_t^i, \boldsymbol{\phi}_t^i] = \sum_{k=1}^{K^i} \phi_{k,t}^i \exp(\beta_{k,t}^{i,\intercal} x_t^i). \tag{4}$$

The proposed generic model (1) enjoys a remarkable property: it gives rise to a number of sub-models of paramount importance. In particular:

1) The *basic Poisson GLM* arises if the mixture degenerates to a single Poisson component:

$$Y_t^i | x_t^i, \beta_t^i \sim \mathit{Pois}\left(\exp(\beta_t^{i,\intercal} x_t^i)\right) \tag{5}$$

with the GLM form

$$\mathbb{E}[Y_t^i | x_t^i, \beta_t^i] = \exp\left(\beta_t^{i,\intercal} x_t^i\right). \tag{6}$$

2) The prominent *zero-inflated Poisson (ZIP) GLM* arises if there are excessive zero measurements unexplained by a single Poisson component:

$$Y_t^i | x_t^i, \beta_t^i, \phi_t^i \sim \phi_t^i \mathit{Dir}(0) + (1 - \phi_t^i) \mathit{Pois}\left(\exp(\beta_t^{i,\intercal} x_t^i)\right), \tag{7}$$

with the corresponding GLM form

$$\mathbb{E}[Y_t^i | x_t^i, \beta_t^i, \phi_t^i] = (1 - \phi_t^i) \exp(\beta_t^{i,\intercal} x_t^i). \tag{8}$$

3) The pure *Poisson GLM mixture models* arise if $K^i > 1$ and the Dirac component is absent,

$$Y_t^i | x_t^i, \boldsymbol{\beta}_t^i, \boldsymbol{\phi}_t^i \sim \sum_{k=1}^{K^i} \phi_{k,t}^i \mathit{Pois}\left(\exp\left(\beta_{k,t}^{i,\intercal} x_t^i\right)\right), \tag{9}$$

where the GLM form is

$$\mathbb{E}[Y_t^i | x_t^i, \boldsymbol{\beta}_t^i, \boldsymbol{\phi}_t^i] = \sum_{k=1}^{K^i} \phi_{k,t}^i \exp(\beta_{k,t}^{i,\intercal} x_t^i). \tag{10}$$

A. *Homogeneity Across the Network*

So far, the situation has been described from the viewpoint of individual network agents that locally acquire measurements y_t^i, regressors x_t^i, and model the process using the ZIPMM (1) or any of its submodels (5), (7), or (9). Based on the operational conditions (sensors technology, their orientation, fields of view etc.) two scenarios may occur:

1) *Homogeneous scenario* where all agents observe the same random process $\{Y_t; t = 1, 2, \ldots\} \equiv \{Y_t^i; t = 1, 2, \ldots\}$ for all t and $i \in \mathcal{I}$. Since Y_t is a random variable, the observations y_t^i may spatially differ, but they still obey the underlying distribution. Similarly, the regressors x_t^i may spatially differ. The ZIPMM model is hence common to all agents $i \in \mathcal{I}$, and they may profit from involving neighbors' measurements and estimates in own inference.
2) *Inhomogeneous scenario* where the ZIPMM may partially or completely differ from agent to agent. Then, the agents may profit from neighbors' estimates, but they need to detect which components are common. An example is depicted in Fig. 1.

We assume that the scenario is known a priori. This allows designing a generic algorithm for collaborative inference in networks with information diffusion. It consists of two steps. During the *adaptation step* the agents update their own prior estimates using the locally measured or shared data. The *combination step* then serves for fusion of estimates that relate to the same mixture components. The steps are described in the sequel.

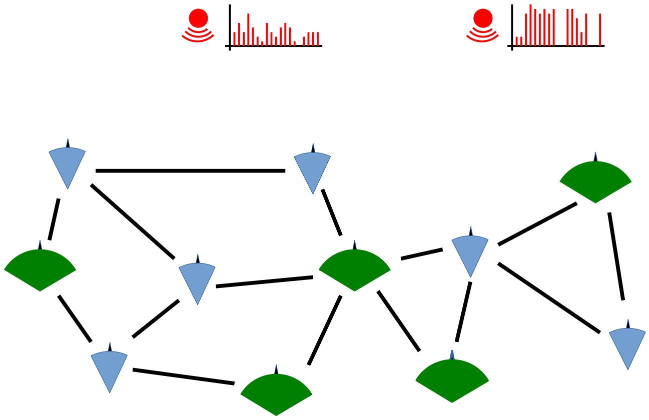

Fig. 1. Example of an inhomogeneous scenario. The network consists of two types of sensors (in blue and green). They have different fields of view and hence observe either one of the processes (in red), or both processes simultaneously.

III. Adaptation Step

The adaptation step serves for the assimilation of available measurements y_t^i and regressors x_t^i into the agents' local knowledge about the inferred variables $\boldsymbol{\beta}_t^i$ and $\boldsymbol{\phi}_t^i$. Recall that these data relate to the same observed process in the homogeneous scenario. Each agent i thus may assimilate y_t^j and x_t^j of neighbors $j \in \mathcal{I}^i$ without harm. By contrast, in the inhomogeneous scenario, the agents observe at least partially different processes, making the use of neighbors' data problematic, if not impossible. The assimilation is hence limited to own data. For convenience, we introduce the adaptation set $\mathcal{I}_A^i$ as follows:

$$\mathcal{I}_A^i = \begin{cases} \mathcal{I}^i & \text{in the homogeneous case,} \\ \{i\} & \text{in the inhomogeneous case.} \end{cases} \tag{11}$$

Let us fix an agent $i \in \mathcal{I}$. With respect to the ZIPMM structure (1), the agent maintains a known number $K^i + 1$ of components, namely one Dirac component and K^i Poisson GLM components. From i's perspective, the observed process activates one concrete component $k_{j,t}^i \in \{0, \ldots, K^i\}$ per each $t = 1, 2, \ldots$ and each $j \in \mathcal{I}_A^i$. This component then produces a measurement y_t^j that is possibly explained by a regressor x_t^j. In particular, either the Dirac distribution is activated ($k_{j,t}^i = 0$) with probability $\phi_{0,t}^i$, or one of the Poisson components is activated ($k_{j,t}^i \in \{1, \ldots, K^i\}$) with probability $\phi_{k,t}^i$. The complete information for *each* measurement $y_t^j, j \in \mathcal{I}_A^i$ can be represented by a joint density

$$f^i(y_t^j, k_{j,t}^i | x_t^j, \boldsymbol{\beta}_t^i, \boldsymbol{\phi}_t^i) = \left[\phi_{0,t}^i \, \delta_0\right]^{\mathbb{1}_0(k_{j,t}^i)} \times \prod_{k=1}^{K^i} \left[\phi_{k,t}^i f_k^i(y_t^i | x_t^i, \beta_{k,t}^i)\right]^{\mathbb{1}_k(k_{j,t}^i)}, \tag{12}$$

where $\mathbb{1}_0(k_{j,t}^i)$ and $\mathbb{1}_k(k_{j,t}^i)$ are the indicators of the Dirac and Poisson components, respectively. They equal 1 if the corresponding component is active at time t, and 0 otherwise. By conditional independence of the information from all the neighbors $j \in \mathcal{I}_A^i$, the complete data model at time t becomes

$$f^i\left(\{y_t^j\}_{j\in\mathcal{I}_A^i}, \{k_{j,t}^i\}_{j\in\mathcal{I}_A^i} \middle| \{x_t^j\}_{j\in\mathcal{I}_A^i}, \boldsymbol{\beta}_t^i, \boldsymbol{\phi}_t^i\right) = \prod_{j\in\mathcal{I}_A^i} f^i(y_t^j, k_{j,t}^i | x_t^j, \boldsymbol{\beta}_t^i, \boldsymbol{\phi}_t^i). \quad (13)$$

This formula explains how the ZIPMM (1) generates data from the ith agent's perspective. We face the ignorance of $\boldsymbol{\phi}_t^i, \boldsymbol{\beta}_t^i$, and $k_{j,t}^i$ whose reliable inference is imperative.

We aim to infer the unknown model parameters from streaming data. For this purpose, we exploit the Bayesian paradigm allowing us to update the knowledge from time $t-1$ to t through the Bayes' theorem. Locally at i, this knowledge is conveyed by the joint prior density

$$\pi^i(\boldsymbol{\beta}_t^i, \boldsymbol{\phi}_t^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i) = \pi^i(\boldsymbol{\beta}_t^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i) \times \pi^i(\boldsymbol{\phi}_t^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i), \quad (14)$$

where X_{t-1}^i, Y_{t-1}^i, and K_{t-1}^i symbolize all knowledge about the past regressors, measurements, and active components indicators available to the ith agent up to time instant $t-1$. Furthermore, by independence of components,

$$\pi^i(\boldsymbol{\beta}_t^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i) = \prod_{k=1}^{K^i} \pi^i(\beta_{k,t}^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i). \quad (15)$$

A. *Local Estimation of* $\boldsymbol{\phi}_t^i$

Assume a fixed agent i. The component weights $\phi_{k,t}^i, k = 0, \ldots, K^i$ take values in the unit K^i-simplex, i.e., they sum to one. This advocates the use of the Dirichlet distribution as the factor for $\boldsymbol{\phi}_t^i$ in the prior distribution (14). The Dirichlet density with the prior hyperparameters $\kappa_{k,t-1}^i \in \mathbb{R}^+$ has the form

$$\pi^i(\boldsymbol{\phi}_t^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i) \propto \prod_{k=0}^{K^i} \left(\phi_{k,t}^i\right)^{\kappa_{k,t-1}^i - 1}, \quad (16)$$

where $\propto$ stands for equality up to the normalizing constant. As new streaming data x_t^j and y_t^j of agents $j \in \mathcal{I}_A^i$ arrive, they are sequentially incorporated into the prior distribution (14) by means of the Bayes' theorem,

$$\pi^i(\boldsymbol{\beta}_t^i, \boldsymbol{\phi}_t^i | X_t^i, Y_t^i, K_t^i) \propto \underbrace{\pi^i(\boldsymbol{\beta}_t^i, \boldsymbol{\phi}_t^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i)}_{\text{prior density (14)}} \underbrace{\prod_{j\in\mathcal{I}_A^i} f^i(y_t^j, k_{j,t}^i | x_t^i, \boldsymbol{\beta}_t^i, \boldsymbol{\phi}_t^i)}_{\text{data model (13)}}. \quad (17)$$

Let us closely focus on individual factors of the formula. If we plug the Dirichlet density (16) into the prior (14), we obtain

$$\pi^i(\boldsymbol{\beta}_t^i, \boldsymbol{\phi}_t^i | \cdot) \propto \pi^i(\boldsymbol{\beta}_t^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i) \times \left(\phi_{0,t}^i\right)^{\kappa_{0,t-1}^i - 1} \prod_{k=1}^{K^i} \left(\phi_{k,t}^i\right)^{\kappa_{k,t-1}^i - 1}. \quad (18)$$

The term $\phi_{0,t}^i$ relates to the Dirac distribution. It is separated for convenience that will become clear shortly. The data model in (17) encompasses the individual agents' densities (12). That is,

$$\prod_{j\in\mathcal{I}_A^i} f^i(y_t^j, k_{j,t}^i | x_t^j, \boldsymbol{\beta}_t^i, \boldsymbol{\phi}_t^i) = \prod_{j\in\mathcal{I}_A^i} \left\{ \left[\phi_{0,t}^i \, \delta_0\right]^{\mathbb{1}_0(k_{j,t}^i)} \cdot \prod_{k=1}^{K^i} \left[\phi_{k,t}^i f_k^i(y_t^j | x_t^j, \beta_{k,t}^i)\right]^{\mathbb{1}_k(k_{j,t}^i)} \right\}. \quad (19)$$

Similarly to the previous density (18), the Dirac-related component is separated.

The compatibility of the prior density (18) and the data models (19) elucidates that the resulting Bayesian update (17) consists of two separate updates. First, the values of the indicators $\mathbb{1}_k(k_{j,t}^i)$ are incorporated into the Dirichlet hyperparameters $\kappa_{k,t-1}^i$. Second, each pair x_t^j and y_t^j updates the relevant factor of the prior distribution for $\boldsymbol{\beta}_t^i$. The major problem is that the formula is purely conceptual, as the indicators of the active components are rarely available. In the sequel, we circumvent this issue by replacing the indicators by their point estimates as suggested by Titterington et al. [37] and Kárný et al. [38].

The Dirichlet density (16) yields the *prior* estimator

$$\mathbb{E}[\phi_{k,t}^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i] = \frac{\kappa_{k,t-1}^i}{\sum_{q=0}^{K^i} \kappa_{q,t-1}^i}. \quad (20)$$

Now, for each Poisson component $k = 1, \ldots, K^i$ and the jth neighbor's data x_t^j, y_t^j, the component indicator point estimators are proportional to the prior probability of the individual components and the predictive likelihood of these components. That is, for $k = 1, \ldots, K^i$,

$$\begin{aligned}\widehat{\mathbb{1}}_k(k_{j,t}^i) &= \mathbb{E}[\mathbb{1}_k(k_{j,t}^i) | X_t^i, Y_t^i, K_t^i] \\ &\propto \mathbb{E}[\phi_{k,t}^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i] \\ &\quad \times f_k^i(y_t^j | x_t^j, X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i). \end{aligned} \quad (21)$$

The Dirac component then takes the rest of the unit probability mass,

$$\begin{aligned}\widehat{\mathbb{1}}_0(k_{j,t}^i) &= \mathbb{E}[\mathbb{1}_0(k_{j,t}^i) | X_t^i, Y_t^i, K_t^i] \\ &= 1 - \sum_{k=1}^{K^i} \mathbb{E}[\mathbb{1}_k(k_{j,t}^i) | X_t^i, Y_t^i, K_t^i] \\ &= 1 - \sum_{k=1}^{K^i} \widehat{\mathbb{1}}_k(k_{j,t}^i), \end{aligned} \quad (22)$$

The Bayesian predictive likelihood of the kth local Poisson component in (21) is

$$f_k^i(y_t^j | x_t^j, X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i) = \int f_k^i(y_t^j | x_t^j, \beta_{k,t}^i) \, \pi^i(\beta_{k,t}^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i) d\beta_{k,t}^i. \quad (23)$$

We will see shortly that this prior predictive likelihood is not analytically tractable. Still, a straightforward way around the problem consists in exploiting the plug-in principle and substitution of the *prior* point estimate $\hat{\beta}_{k,t}^i$ directly into the Poisson density (3),

$$f_k^i(y_t^j|x_t^j, X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i) \approx f_k^i(y_t^j|x_t^j, \hat{\beta}_{k,t}^i). \tag{24}$$

The estimator $\hat{\beta}_{k,t}^i$ will be derived in the following Section III-B.

The same process applies to the Dirac component. The situation is trivial there as it covers only the cases $y_t^j = 0$. To conclude, the Bayesian estimation formulas for $\boldsymbol{\phi}_t^i$ result from (17), (21) and (22). For the Dirac component weight $\phi_{0,t}^i$ the update of the relevant Dirichlet hyperparameter reads

$$\kappa_{0,t}^i = \kappa_{0,t-1}^i + \sum_{j \in \mathcal{I}_A^i} \widehat{\mathbb{1}}_0(k_{j,t}^i). \tag{25}$$

For the Poisson component weights $\phi_{1,t}^i, \ldots, \phi_{K^i,t}^i$, the update formula is

$$\kappa_{k,t}^i = \kappa_{k,t-1}^i + \sum_{j \in \mathcal{I}_A^i} \widehat{\mathbb{1}}_k(k_{j,t}^i), \qquad k = 1, \ldots, K^i. \tag{26}$$

Remark 1: While using the parameter point estimate directly in the generative model is standard in the frequentist statistic, its use in the Bayesian analyses is often considered controversial. This is due to the fact that the predictive distributions of the form (23) account for uncertainty about $\beta_{k,t}^i$. In practice, the trade-off between uncertainty quantification and computational burden is often acceptable if the sample size is large enough. Moreover, Smith pointed out that the inferiority of the plug-in estimators is by far not a rule [39].

B. Approximate Local Estimation of $\boldsymbol{\beta}_t^i$

From the independence of $\beta_{k,t}^i$ it follows that the Bayesian update (17) with the estimated indicators (21) results in the posterior distribution of $\boldsymbol{\beta}_t^i$ of the form

$$\begin{aligned}\pi^i(\boldsymbol{\beta}_t^i|X_t^i, Y_t^i, K_t^i) &= \prod_{k=1}^{K^i} \pi^i(\beta_{k,t}^i|X_t^i, Y_t^i, K_t^i) \\ &\propto \prod_{k=1}^{K^i} \pi^i(\beta_{k,t}^i|X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i) \prod_{j \in \mathcal{I}_A^i} \left[f_k^i(y_t^j|x_t^j, \beta_{k,t}^i) \right]^{\widehat{\mathbb{1}}_k(k_{j,t}^i)}.\end{aligned} \tag{27}$$

A careful examination of this update elucidates that the posterior distribution of each $\beta_{k,t}^i$ incorporating the measurement y_t^j and the regressor x_t^j is given by the weighted Bayesian update

$$\pi^i(\beta_{k,t}^i|X_t^i, Y_t^i, K_t^i) \propto \pi^i(\beta_{k,t}^i|X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i) \times \prod_{j \in \mathcal{I}_A^i} \left[f_k^i(y_t^j|x_t^j, \beta_{k,t}^i) \right]^{\widehat{\mathbb{1}}_k(k_{j,t}^i)}. \tag{28}$$

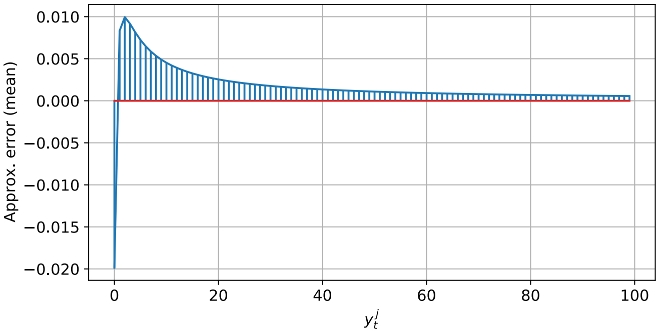

Fig. 2. Mean value $m_{k,t}^j$: The approximation error (residue) for different values of y_t^j. The order of magnitude is very low.

In principle, this update can be performed in a sequential one-by-one way. The problem of its analytical intractability can theoretically be overcome using the Bartlett and Kendall's approximation [40]. Namely, for two random variables a and b with a density $p(a|b)$,

$$p(a|b) \propto e^{ab} e^{-exp(a)} \Rightarrow a \mathrel{\dot{\sim}} \mathcal{N}\left(\log b, \frac{1}{b}\right). \tag{29}$$

This approach was first used for static Poisson GLM by El Sayaad [41]. We aim at applying this Gaussian approximation to the Poisson components. Indeed, their densities (3) are compatible with (29). From a practical viewpoint, this would work well for y_t^i large enough. For low values, the approximation is crude, and it inevitably fails if $y_t^j = 0$. Inspired by [42] and [27], we rewrite the density (3) as follows:

$$\begin{aligned}f_k^i(y_t^j|x_t^j, \beta_{k,t}^i) &= \frac{\exp(\beta_{k,t}^{i,\intercal} x_t^j (y_t^j + 1) - \exp(\beta_{k,t}^{j,\intercal} x_t^j))}{y_t^j!} \\ &\quad \times \frac{1}{\exp(\beta_{k,t}^{i,\intercal} x_t^j)}.\end{aligned} \tag{30}$$

This form is stable under $y_t^j = 0$ and its first factor is compatible with (29). However, the approximation is still relatively crude for very low values of y_t^j. The difference between the original mean value and the mean value of the approximating Gaussian distribution is high for low y_t^j, but quickly decreases with increasing y_t^j. The same is true for the standard deviation. In order to suppress these errors for all values of y_t^j, we fit a simple hyperbolic regression model describing the evolution of the pair "$y_t^j \sim$ error in mean," and another hyperbolic regression model for the pair "$y_t^j \sim$ error in standard deviation". The predicted errors are then subtracted from the mean and standard deviation of the original approximation (29). The resulting calibrated Gaussian approximation is convenient for any value of y_t^j. It is given by $\mathcal{N}(m_{k,t}^j, s_{k,t}^{j,2})$ where

$$m_{k,t}^j = \log(y_t^j + 1) - \frac{0.5574}{y_t^j + 1}, \tag{31a}$$

$$s_{k,t}^j = \frac{1}{\sqrt{y_t^j + 1}} + \frac{0.0724}{y_t^j + 1} + \frac{0.2121}{(y_t^j + 1)^2}. \tag{31b}$$

The approximation accuracy present Figs. 2, 3, and 4. The first two depict the final error for the mean value (31a) and the standard deviation (31b), respectively. Apparently, the orders

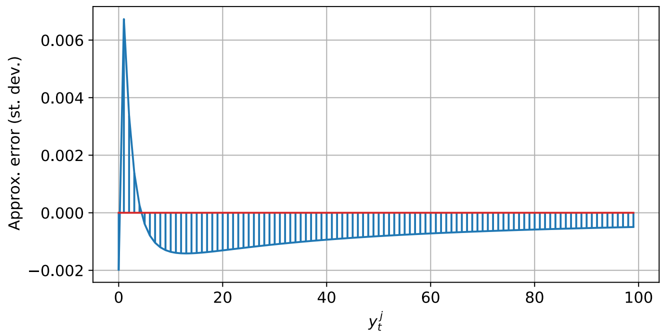

Fig. 3. Standard deviation $s_{k,t}^j$: The approximation error for different values of y_t^j. The order of magnitude is remarkably low.

of inaccuracy are negligible. Fig. 4 compares the cumulative distribution functions and the Q-Q plots of the true distribution, the original crude approximation (29), and the calibrated approximation (31). The error apparently vanishes with an increasing value of y_t^j regardless of the approximation type. For low values, the calibration leads to much better results.

The calibrated approximation of the data model (30) results from the normal density with the mean and standard deviation (31a) and (31b), and the second factor in (30). Its basic form and the exponential family form characterized by the sufficient statistic read

$$f_k^i(y_t^j|x_t^j,\beta_{k,t}^i) \propto \exp\left(\frac{-1}{2s_{k,t}^{j,2}}(\beta_{k,t}^{i,\intercal}x_t^j - m_{k,t}^j)^2 - \beta_{k,t}^{i,\intercal}x_t^j\right)$$

$$\propto \exp\left\{\frac{-1}{2}\,\mathrm{Tr}\left(\begin{bmatrix}-1\\ \beta_{k,t}^i\end{bmatrix}\begin{bmatrix}-1\\ \beta_{k,t}^i\end{bmatrix}^\intercal T(x_t^j,y_t^j,k_{j,t}^i)\right)\right\}, \tag{32}$$

respectively. The sufficient statistic $T(x_t^j,y_t^j,k_{j,t}^i)$ is a square symmetric $(n+1)\times(n+1)$ matrix

$$T(x_t^j,y_t^j,k_{j,t}^i) = \frac{1}{s_{k,t}^{j,2}}\begin{bmatrix} m_{k,t}^{j,2} & (m_{k,t}^j - s_{k,t}^{j,2})x_t^{j,\intercal} \\ x_t^j(m_{k,t}^j - s_{k,t}^{j,2}) & x_t^j x_t^{j,\intercal}\end{bmatrix}. \tag{33}$$

Let us again focus on the update (28) where we substitute the Gaussian densities (32) for the data models $f_k^i(y_t^j|x_t^j,\beta_{k,t}^i)$. In order to make the update analytically tractable, the Bayesian paradigm requires *conjugacy* between the data models and the prior $\pi^i(\beta_{k,t}^i|X_{t-1}^i,Y_{t-1}^i,K_{t-1}^i)$. A convenient choice is the Gaussian density

$$\pi^i(\beta_{k,t}^i|X_{t-1}^i,Y_{t-1}^i,K_{t-1}^i) \equiv \mathcal{N}(b_{k,t-1}^i, P_{k,t-1}^i), \tag{34}$$

where $b_{k,t-1}^i \in \mathbb{R}^n$ is the mean vector and $P_{k,t-1}^i$ is the $n\times n$ positive semi-definite covariance matrices. Its probability density reads

$$\begin{aligned}&\pi^i(\beta_{k,t}^i|X_{t-1}^i,Y_{t-1}^i,K_{t-1}^i)\\ &\quad\propto \exp\left(-\frac{1}{2}(\beta_{k,t}^i - b_{k,t-1}^i)^\intercal P_{k,t-1}^{i,-1}(\beta_{k,t}^i - b_{k,t-1}^i)\right)\\ &\quad\propto \exp\left\{-\frac{1}{2}\,\mathrm{Tr}\left(\begin{bmatrix}-1\\ \beta_{k,t}^i\end{bmatrix}\begin{bmatrix}-1\\ \beta_{k,t}^i\end{bmatrix}^\intercal \Psi_{k,t-1}^i\right)\right\},\end{aligned} \tag{35}$$

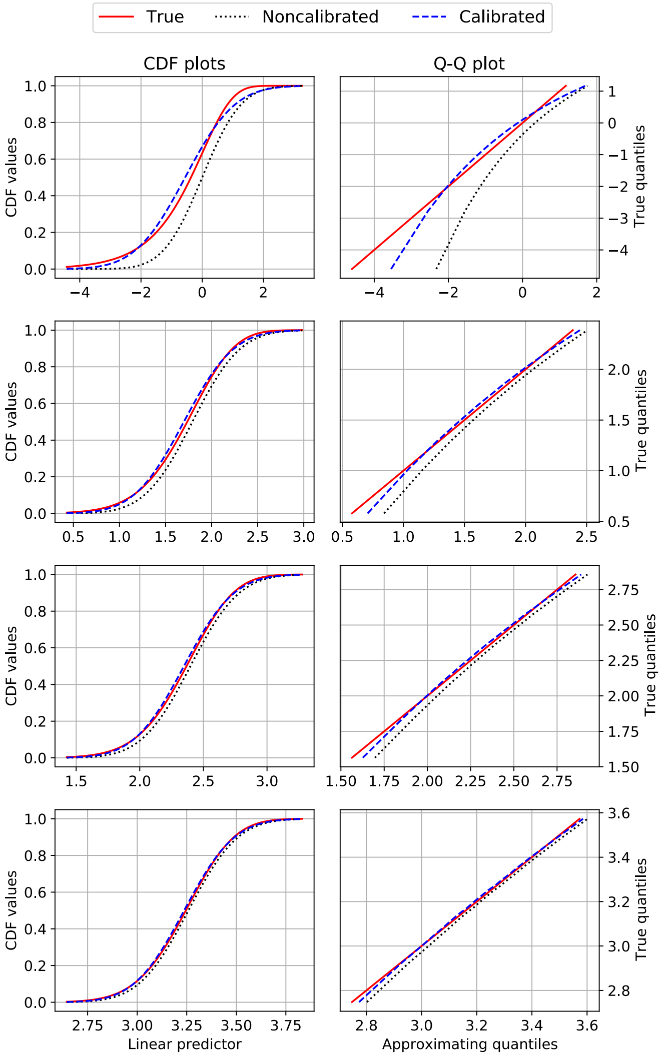

Fig. 4. Quality of the calibrated and original approximations of the true density for y_t^j taking values 1, 5, 10, and 25 (rows from the top). The calibration significantly improves the approximation quality for lower values.

where

$$\Psi_{k,t-1}^i = \begin{bmatrix} b_{k,t-1}^{i,\intercal} \\ I\end{bmatrix} P_{k,t-1}^{i,-1}\begin{bmatrix} b_{k,t-1}^{i,\intercal} \\ I\end{bmatrix}^\intercal \tag{36}$$

is the $(n+1)\times(n+1)$ symmetric sufficient statistic, and I is an $n\times n$ identity matrix.

As a result, the Bayesian update (28) multiplies the prior distribution (35) and a product of the data models (32) weighted by estimated component indicators $\widehat{\mathbb{1}}_k(k_{j,t}^i)$. Since the densities with sufficient statistics are functionally compatible, the update assimilates the sufficient statistics $T(x_t^j,y_t^j,k_{j,t}^i)$ into $\Psi_{k,t-1}^i, k=1,\ldots,K^i$ as follows:

$$\begin{aligned}\Psi_{k,t}^i &= \Psi_{k,t-1}^i + \sum_{j\in\mathcal{I}_A^i}\widehat{\mathbb{1}}_k(k_{j,t}^i)T(x_t^j,y_t^j,k_{j,t}^i)\\ &= \begin{bmatrix}(\psi_{k,t}^i)_{11} & (\psi_{k,t}^i)_{12}\\ (\psi_{k,t}^i)_{12}^\intercal & (\psi_{k,t}^i)_{22}\end{bmatrix}.\end{aligned} \tag{37}$$

The term $(\psi_{k,t}^i)_{11}$ is scalar, and the submatrix $(\psi_{k,t}^i)_{2,2}$ has the same dimension as $P_{k,t}^{i,-1}$. The posterior $\Psi_{k,t}^i$ then easily yields

$$\begin{aligned} P_{k,t}^i &= (\psi_{k,t}^i)_{2,2}^{-1}, \\ b_{k,t}^i &= (\psi_{k,t}^i)_{2,2}^{-1} (\psi_{k,t}^i)_{1,2}^{\intercal}. \end{aligned} \tag{38}$$

There are certain attractive features of working directly with $\Psi_{k,t}^i$ in place of the original $b_{k,t}^i$ and $P_{k,t}^i$. First, the additive form of the update (37) avoids potentially computationally unstable matrix inversions. Second, it represents all available information about $\beta_{k,t}^i$ in a form allowing a straightforward information fusion in networks. We will exploit this property in Section IV. And third, it is easy to discount potentially outdated information about time-varying parameters. The following section covers this.

C. Time-Varying Parameters $\boldsymbol{\beta}_t^i$ and $\boldsymbol{\phi}_t^i$

In most real-world applications, the model parameters are not constant, yet there is no known evolution model. A popular way around the problem is to assume that they follow a random walk. If the noise power of this random walk is smaller than the noise power of the measurement noise, we say that the parameters vary slowly. Then, it is possible to build an adaptive filter that can track the parameters [43, Chap. 16]. The favored approach is to gradually discount older information from the previous posterior distribution before its subsequent updating [44], [45]. This amounts to exponentiation of the density using forgetting factors ranging from 0.95 (fast forgetting) to 1 (no forgetting). The concrete values should balance the trade-off between tracking and noise sensitivity. In practice, they are selected heuristically.

Forgetting factor α_β applied to the estimator of $\beta_{k,t}^i$ yields

$$\pi^i(\beta_{k,t}^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i) = [\pi^i(\beta_{k,t-1}^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i)]^{\alpha_\beta}, \tag{39}$$

which obviously deflates $\Psi_{k,t-1}^i$ and hence reduces the amount of information in it,

$$\Psi_{k,t-1}^i \leftarrow \alpha_\beta \Psi_{k,t-1}^i \quad \forall k \in \{1, \ldots, K^i\}. \tag{40}$$

Similarly, for ϕ_t^i and forgetting factor α_ϕ,

$$\pi^i(\phi_t^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i) = [\pi^i(\phi_{t-1}^i | X_{t-1}^i, Y_{t-1}^i, K_{t-1}^i)]^{\alpha_\phi}, \tag{41}$$

which deflates the Dirichlet hyperparameters $\kappa_{k,t-1}^i$,

$$\kappa_{k,t-1}^i \leftarrow \alpha_\phi \kappa_{k,t-1}^i \quad \forall k \in \{0, \ldots, K^i\}. \tag{42}$$

The resulting hyperparameters subsequently enter the Bayesian updating steps described earlier.

IV. Combination Step

After the adaptation step, the network agents $i \in \mathcal{I}$ possess the posterior estimates $\pi^i(\boldsymbol{\beta}_t^i|\cdot)$. With respect to the networked environment, there arises a logical (if not instinctive) impetus for taking advantage of neighbors' knowledge. Inspired by human behavior, we suggest that an agent should consider only knowledge that is sufficiently similar to its own. This principle can be applied to both the homogeneous and inhomogeneous case. In the former, it primarily protects from erroneous or poisoned information. In the second case, it facilitates the identification of common components.

In order to develop the combination step, let us fix an agent $i \in \mathcal{I}$. Its estimators of unknown vectors $\beta_{k,t}^i, k = 1, \ldots, K^i$ are its local posterior densities $\pi^i(\beta_{k,t}^i|\cdot)$. Their mean vectors $b_{k,t}^i$ stand for the point estimates, while their covariances $P_{k,t}^i$ express the associated uncertainty. The same principle applies to i's neighbors $j \in \mathcal{I}^i$. Recall, that i has access to their posteriors, some of which may relate to vectors $\beta_{k,t}^i, k = 1, \ldots, K^i$. Their identification and subsequent fusion could significantly improve i's statistical knowledge about the inferred quantities. The identification is based on the similarity of point estimates. For each *locally* inferred $\beta_{k,t}^i$, each i's neighbor $j \in \mathcal{I}^i$ with posterior densities $\pi^j(\beta_{l,t}^j | X_t^j, Y_t^j, K_t^j), l \in \{1, \ldots, K^j\}$, we define the set $\mathcal{I}_C^i(\beta_{k,t}^i)$ of similar densities. It consists of elements yielding point estimates that lie within a ball of radius $d \geq 0$ around $b_{k,t}^i$:

$$\mathcal{I}_C^i\left(\beta_{k,t}^i\right) = \left\{\pi^j(\beta_{l,t}^j|\cdot) : \left|\left|b_{l,t}^j, b_{k,t}^i\right|\right|_2 \leq d, \forall j \in \mathcal{I}^i\right\}. \tag{43}$$

If $d = 0$, the sets would contain only i's own densities and no neighbors' ones. As the value of d increases, the set becomes gradually richer. However, if d becomes too large, the set may contain even poisoned information or information relevant to other components. Appropriate setting of d is problem-specific and depends on the observed system. For instance, the user may set it based on the (rough) expertise or preliminary analyses of how distant the inferred vectors are. It is also possible to start with a very conservative (low) value of d and increase it when the posterior distributions become sufficiently concentrated. Alternatively, it is possible to run a bank of estimators with different values of d and gradually select those that yield the best modeling performance.

Remark 2: It may be argued that the selection of similar densities should consider the covariance matrices too. Various information divergences such as the Kullback-Leibler or the Bregman divergence could be used for this task. However, in our Bayesian information processing, the densities express the location of the point estimate (the mean) and the uncertainty about this location (the covariance matrix). The agents are primarily interested in the former, while the latter is a penalizing factor for the subsequent analyses.

The rule for the local combination of the elements of $\mathcal{I}_C^i(\beta_{k,t}^i)$ should respect the fundamental principles of Bayesian information processing. By the Fisher-Neyman theorem, the parameter estimators depend on data only through the sufficient statistics [46]. These statistics are assimilated to the additive hyperparameter $\Psi_{k,t}^i$, cf. (37). By the Bernstein-von Misses theorem, the influence of the initial $\Psi_{k,0}^i$ on the posterior distribution vanishes with $t \to \infty$ under a well-specified parametric model with a finite number of parameters [46]. That is, the dominating information is contained in the sum of sufficient statistics. The intended combination rule should protect this additivity and ensure that if the same distributions were combined, the result would stay identical, i.e., there would be no information loss nor gain. The arithmetic mean of the posterior hyperparameters

Algorithm 1: Robust Online Modeling of Counts in Agent Networks.

```
For each agent $i \in \mathcal{I}$ set the initial hyperparameters $\Psi^i_{k,0}$
  and $\kappa^i_{k,0}$ of components $k = 1, \ldots, K^i$, and $\kappa^i_{0,0}$ of the
  Dirac component. Set the forgetting factors $\alpha_\phi$ and $\alpha_\beta$.
  Set the acceptable deviation $d$.
For $t = 1, 2, \ldots$ and each agent $i \in \mathcal{I}$ do:
Adaptation step:
    1) Perform forgetting, (40) and (42).
    2) Acquire $x^j_t, y^j_t$ of neighbors $j \in \mathcal{I}^i_A$, cf. (11)
    3) Calculate the expected component indicators $\widehat{\mathbb{1}}_k(k^i_{j,t})$
       and $\widehat{\mathbb{1}}_1(k^i_{j,t})$, (21), (22), and (23).
    4) Update the prior for $\boldsymbol{\phi}^i_t$, (25) and (26).
    5) Calculate the sufficient statistics $T(x^j_t, y^j_t, k^i_{j,t})$, (33),
       using moments (31).
    6) Update the prior for $\boldsymbol{\beta}^i_t$, (37).
Combination:
    1) Acquire poster. densities $\pi^j(\boldsymbol{\beta}^j_t|\cdot)$ of neighbors $j \in \mathcal{I}^i$.
    2) Construct the sets $\mathcal{I}^i_C(\cdot)$ of similar components, (43).
    3) Combine similar components, (44).
Estimation: Calculate the point estimates of $\boldsymbol{\phi}^i_t$ and
  $\beta^i_{k,t}, k = 1, \ldots, K^i$, (20) and (38).
```

resulting in the combined posterior density

$$\bar{\pi}^i(\beta^i_{k,t}|\cdot) \propto \exp\left\{\frac{-1}{2}\mathrm{Tr}\left(\begin{bmatrix}-1\\ \beta^i_{k,t}\end{bmatrix}\begin{bmatrix}-1\\ \beta^i_{k,t}\end{bmatrix}^{\intercal}\frac{\sum_{j\in\mathcal{I}^i_C(\beta^i_{k,t})}\Psi^j_{k,t}}{\mathrm{card}\{\mathcal{I}^i_C(\beta^i_{k,t})\}}\right)\right\} \tag{44}$$

fulfills the requirement, cf. (37). The mean of the resulting distribution serves as the point estimator of $\beta^i_{k,t}$.

Remark 3: The combination rule is widely used in information theory and statistics. It can be shown that it is Kullback-Leibler-optimal [47] and Bregman-optimal [48]. The same rule can be used to combine information about $\boldsymbol{\phi}^i_t$ if the scenario is totally homogeneous and all agents observe *exactly* the same process. We leave it beyond the paper scope for its relatively limited use.

V. Properties

a) Communication Requirements: The communication requirements of the adaptation step at each agent $i \in \mathcal{I}$ amounts to card $(\mathcal{I}^i_A)$ n-dimensional real vectors x^j_t and the same number of nonnegative integer measurements y^j_t, $j \in \mathcal{I}^i_A$. The communication burden of the combination step at an agent i involves K^j *symmetric* real matrices $\Psi^j_{k,t}$ of the dimension $(n+1) \times (n+1)$ to be transmitted for each neighbor $j \in \mathcal{I}^i$.

b) Memory Requirements: The memory requirements are as follows: The local prior density for $\boldsymbol{\phi}^i_t$ requires $(K^i + 1)$ floating point numbers $\kappa^i_{0,t}, \ldots, \kappa^i_{K^i,t}$. The local prior density for $\boldsymbol{\beta}^i_t$ requires K^i symmetric $(n+1) \times (n+1)$ matrices $\Phi^i_{k,t}$ of floating point numbers. The adaptation step requires memory for card$(\mathcal{I}^i_A)$ integer measurements y^j_t, and memory for card$(\mathcal{I}^i_A)$ regression vectors x^j_t of n floating point numbers. During the combination step, each neighbor $j \in \mathcal{I}^i$ provides K^j symmetric matrices $\Psi^j_{k,t}$ of $(n+1) \times (n+1)$ floating point numbers. The algorithm requires setting of three floating point constants: $\alpha_\phi, \alpha_\beta$, and d. Since the prior distributions are conjugate, all information necessary for estimation is stored in sufficient statistics of fixed size. No variable changes its size in time.

c) Computational Complexity: Unlike the existing solutions relying mostly on expectation maximization or MCMC methods (see Section I), the proposed algorithm has a fully sequential *noniterative* character. The number of arithmetic operations depends on the number of modeled components and the cardinality of agent's neighborhood. The use of sufficient statistics reduces the Bayesian updates to weighted summations of (i) $K^i + 1$ floating point numbers – (25), (26), and (ii) symmetric $(n+1) \times (n+1)$ matrices (37). Similarly, the combination step (44) is a summation of symmetric $(n+1) \times (n+1)$ matrices. Matrix inversion is used only in the evaluation of the covariance matrix $P^i_{k,t-1}$ in (38). However, the algorithm does not need its calculation *per se*. There is a need to evaluate the (scalar) value of the Gaussian density during the adaptation step in (24) and the (scalar) moments $m^j_{k,t}, s^j_{k,t}$ in Formulas (31). One local time step of Algorithm 1 takes several milliseconds on a desktop computer if a naive implementation is written in `python` with `numpy`. An optimized implementation would allow to use the method in relatively fast real-time applications.

d) Limitations: The limitations of the proposed algorithm are connected with the equidispersion assumption: The variance of a *single-component* Poisson variable is identical to the mean value. If the single-component variable exhibits significant overdispersion, i.e., the variance exceeds the mean, some other count models are preferred, namely the negative binomial, Poisson-inverse-Gaussian, or the generalized Poisson model. In the case of underdispersion, the generalized Poisson model can be used too [49]. Another very flexible possibility is the Conway-Maxwell-Poisson model [50]. However, these models, if used as GLMs, share a common difficulty – their estimation is (in some cases extremely) computational demanding. This prevents their use for online sequential modeling from streaming data. We plan to focus on this in future work. We emphasize that the present paper primarily assumes overdispersion due to multiple sub-processes that generate the data. The corresponding data model is thus a multi-component mixture.

VI. Illustrative Examples

The aim of the following two examples is twofold. First, they empirically study the properties of the proposed algorithm, as the involved approximations hinder its theoretical analyses. Second, they demonstrate the performance of the algorithm. The first example focuses on *homogeneous* networks where all agents observe the same stationary process with constant parameters. The aim is to demonstrate that the estimates converge to the true values and that the collaboration among agents accelerates this convergence. The second example considers an *inhomogeneous* network of possibly differently operating sensors. There exist two count processes, and the agents observe either one or both of

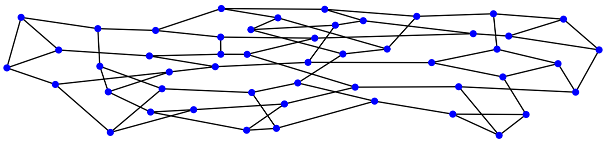

Fig. 5. Example 1: Network topology.

TABLE I
EXAMPLE 1: FINAL VALUES OF RMSE AVERAGED OVER THE NETWORK AND 100 EXPERIMENT RUNS

	MLE	**NOCOOP**	**ATC**
β **estimates**	0.0161	0.0163	0.0027
ϕ **estimates**	0.0235	0.0221	0.0116

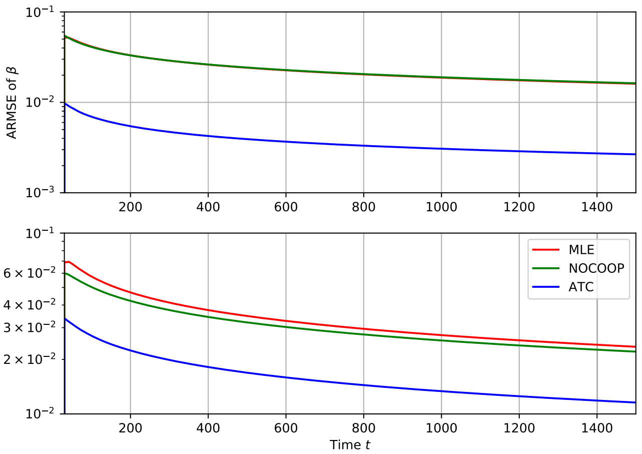

Fig. 6. Example 1: Evolution of RMSE averaged over the network and 100 experiment runs.

them. Their measurements are locally prone to excessive zeros imitating the dead times. The component weights thus differ from agent to agent. In addition, the parameters are time-varying. The goal is to study the robustness and performance of the algorithm.

A. Example 1: Homogeneous Network

In this example, we assume a randomly generated homogeneous network consisting of 50 agents. Its topology is depicted in Fig. 5. The simulated data-generating system consists of one common Dirac component $Dir(0)$ with weight $\phi_{0,t}^i = 0.3$, and one common Poisson component $Pois(\exp(\beta_{1,t}^{i,\intercal} x_t^i))$ with $\phi_{1,t}^i = 0.7$ for all $i \in \mathcal{I}$. The global vector $\beta_{1,t}^i = [0.1, 0.2, 0.3]^\intercal$. The regressors $x_t^i \sim U(0,5)^3$ are simulated independently for each agent. This setting is fixed for all $t = 1, \ldots, 1500$. As the parameters are constant, the forgetting factors $\alpha_\beta = \alpha_\phi = 1$.

The initial prior distributions of all $i \in \mathcal{I}$ are set as follows: The Dirichlet distribution hyperparameters $\phi_{0,t}^i = \phi_{1,t}^i = 1$. The normal distribution hyperparameters used for constructing $\Psi_{k,0}^i$ in Formula (36) are $b_{1,0}^i = [0,0,0]^\intercal$ and $P_{1,0}^i = 100 \cdot I$ where I is a 3×3 identity matrix.

The following strategies are compared: (i) NOCOOP, where the agents do not collaborate at all, (ii) ATC, where the adapt-then-combine Algorithm 1 is used, and (iii) MLE corresponding to the maximum likelihood-based estimation [29]. The MLE strategy illustrates the performance of the standard state-of-the-art approach to the ZIPMM estimation problem. This strategy is noncollaborative for obvious reasons, hence it corresponds to the frequentist variant of NOCOOP. Furthermore, as MLE does not operate with any prior knowledge, it requires sufficient amount of initial data to avoid numerical issues. Therefore, the analyses are evaluated for $t \geq 30$. The results are averaged over 100 independent experiment runs.

The estimation performance in terms of the root mean square error (RMSE) averaged over the agents and runs is depicted in Fig. 6. Table I shows numerical comparison of the final values.

Apparently, the collaboration among agents significantly accelerates the estimator convergence, while the noncollaborative strategies NOCOOP and MLE exhibit similar behavior. NOCOOP and MLE provide almost identical estimates of $\boldsymbol{\beta}$, and NOCOOP performs only negligibly better in the estimation of $\boldsymbol{\phi}$. The estimation theory dictates that the Bayesian and MLE estimates are asymptotically equivalent [51], and this empirically proves validity of the proposed estimator. It is critical to emphasize that the state-of-the-art MLE is not capable of sequential (online) updating. Instead, the estimates are calculated at each time instant from scratch on an expanding data window. This is *extremely* computationally and memory-intensive.

Finally, Fig. 7 depicts the evolution of point estimates for a randomly chosen run at a randomly selected agent. It is obvious that the estimates gradually converge with an increasing amount of incorporated data. As the ATC strategy diffuses the information about the inferred quantities through the network, the convergence is effectively accelerated and more stable.

B. Example 2: Inhomogeneous Network

The second example considers the same network (Fig. 5). It studies the estimator performance under significantly relaxed conditions. The process is constructed as follows: There exist two count processes with the vectors of regression coefficients

$$\beta_{1,t} = \begin{bmatrix} 0.05 + 0.03 \cdot \sin\left(\frac{t}{1500}\pi\right) \\ 0.10 + 0.03 \cdot \sin\left(\frac{t}{1500}\pi\right) \\ 0.15 + 0.03 \cdot \sin\left(\frac{t}{1500}\pi\right) \end{bmatrix} \tag{45}$$

and

$$\beta_{2,t} = \begin{bmatrix} 0.30 - 0.03 \cdot \sin\left(\frac{2t}{1500}\pi\right) \\ 0.35 - 0.03 \cdot \sin\left(\frac{2t}{1500}\pi\right) \\ 0.40 - 0.03 \cdot \sin\left(\frac{2t}{1500}\pi\right) \end{bmatrix}, \tag{46}$$

respectively. The time instants $t = 1, \ldots, 1500$. The regressors $x_t^i \sim U(0,5)^3$ are generated independently for each $i \in \mathcal{I}$. The two processes mimic two physically different phenomena, e.g., spatially separated or with different wavelengths. Some agents can observe only one of the two components, and some can observe both components. This corresponds to different fields of views or to different measuring principles. In addition, each agent is subject to excessive zeros due to dead times. All

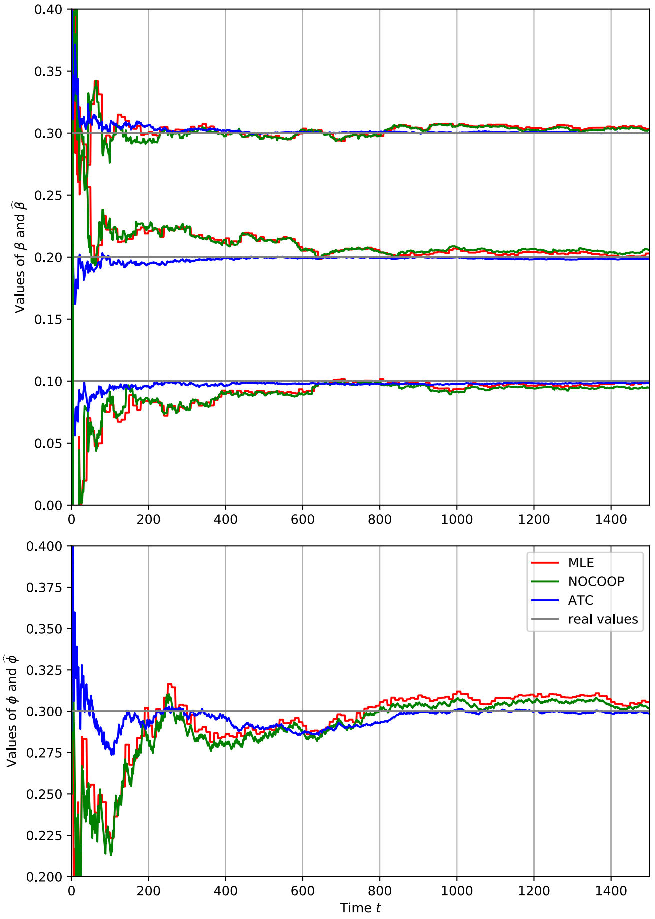

Fig. 7. Example 1: Evolution of estimates at a randomly chosen agent and run.

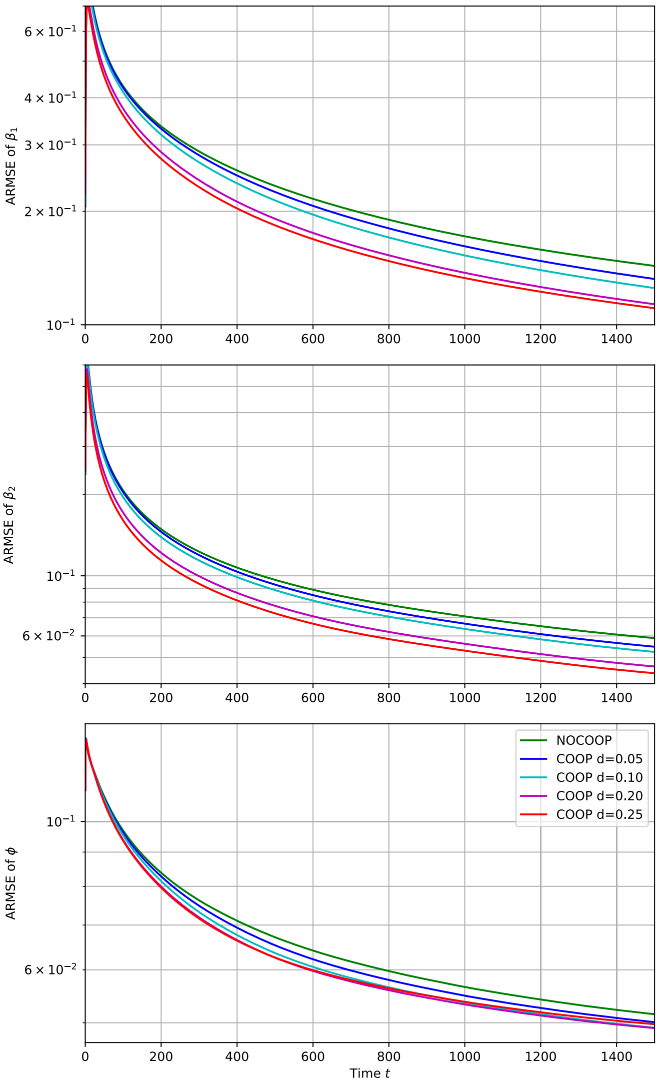

Fig. 8. Example 2: Evolution of RMSE averaged over the network and 100 experiment runs.

parameters are slowly time-varying, and the forgetting factors $\alpha_\phi = \alpha_\beta = 0.99$ are adopted.

The agents' prior knowledge is limited only to the number of components. They are not aware of the total number of components in the network, nor which of the neighbors observe the same Poisson component(s). If an agent observes just one Poisson component, the hyperparameter $\Psi^i_{1,t}$ in (33) is constructed with $b^i_{1,0} = [0, 0, 0]^\intercal$ and $P^i_{1,0} = 100 \cdot I$ where I is a 3×3 identity matrix. In the case of two observed Poisson components, $b^i_{1,0} = [0, 0, 0]^\intercal, b^i_{2,0} = [0.5, 0.5, 0.5]^\intercal, P^i_{1,0} = P^i_{2,0} = 100 \cdot I$. The Dirichlet prior for weights has $\kappa^i_{k,0} = 1$ for all $k = 0, \ldots, K^i$.

Basically, two scenarios are considered. First, the NOCOOP scenario, where the agents do not cooperate at all. Second, the ATC scenario with d equal to $0.05, 0.1, 0.2$, and 0.25. Since the MLE approach assumes constant parameters, it is not included. The results are averaged over 100 independent experiment runs.

Figs. 8 and 9 depict the achieved results. Again, let us emphasize that the parameters are not constant. The optimal solution is no longer fixed and the algorithm is required to continually track its variations. The algorithm must first pass through a *transient phase* in order to converge sufficiently close to the true parameter values. Then, it continually adjusts the estimates values during the tracking. Tracking is hence a *steady state* phenomenon [43]. From the figures follows that the proposed estimator is able to initially converge and subsequently track the parameters variations. The performance is connected with the level of collaboration which is driven by d. Namely, in estimation of $\boldsymbol{\beta}^i_t$, the noncollaborative scenario (equivalent to $d = 0$) performs significantly worse than the collaborative scenarios ($d > 0$). Under collaboration, the agents effectively estimate from richer amount of information. In particular, Fig. 9 (top) demonstrates that the estimates are much smoother and close to the true values of the regression coefficients if the agents collaborate. Otherwise, the estimation quality is rather poor. Recall, that there is no exchange of information about the component weights $\boldsymbol{\phi}^i_t$, as they are *strictly local*. Figs. 8 (bottom) and 9 (bottom) indicate that the estimation performance is practically identical for both the COOP and NOCOOP scenarios, the differences of average RMSE are of order 10^{-3}. In contrast to the constant-parameters case in Example 1, tracking of the component weights is very challenging for both COOP and NOCOOP scenarios.

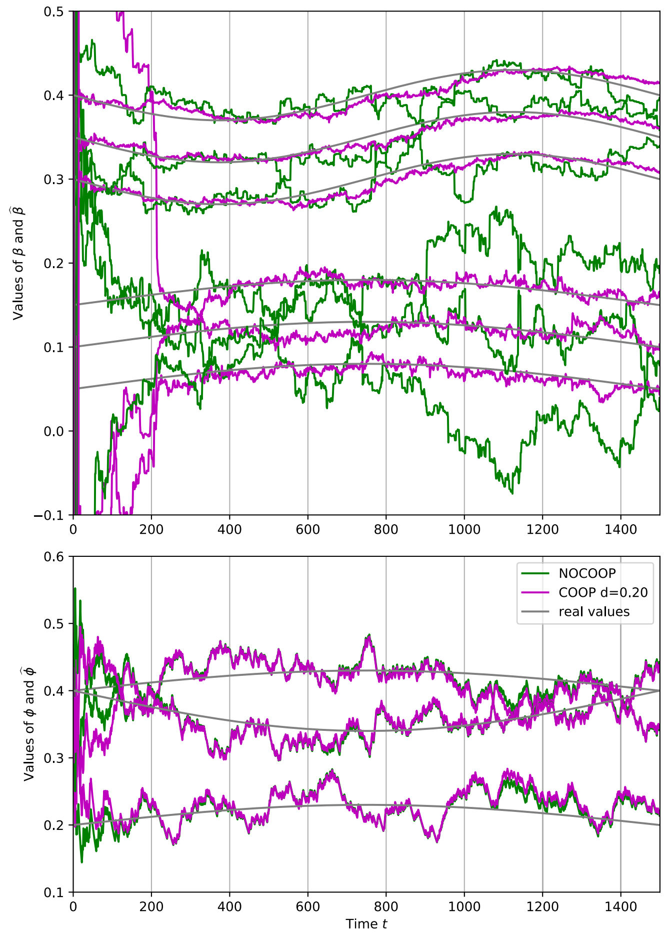

Fig. 9. Example 2: Evolution of estimates at a randomly chosen agent and run.

To summarize, the situations where the model parameters vary are highly sensitive to the amount of available information. The collaboration among agents is a decisive factor of the model inference quality. In our particular case of the ZIPMM, the collaboration allows for stable tracking of the regression coefficients β_t^i.

VII. Conclusion

In this paper, we investigated the collaborative online inference of the zero-inflated Poisson (mixture) models from streaming data. The simulation examples demonstrate that the quality of estimates corresponds to the traditional offline maximum likelihood estimates if the network agents do not collaborate and the parameters are constant. The collaboration of agents significantly accelerates the convergence and stabilizes the estimates close to the true parameter values even if they slowly vary in time. In addition, the algorithm accounts for network inhomogeneity, where the agents possibly observe partially or completely different processes. Unlike the existing solutions, the proposed algorithm exploits additive sufficient statistics and hyperparameters, allowing for fully sequential computations. It does not rely on computationally expensive iterative procedures nor Monte Carlo methods. Future work can concentrate on unknown numbers of components, or models of underdispersed and overdispersed variables. For instance, the beta-binomial or the very flexible but challenging Conway-Maxwell-Poisson models seem to be very attractive candidates.

References

[1] A. H. Sayed, “Adaptation, learning, and optimization over networks,” *Found. Trends Mach. Learn.*, vol. 7, no. 4/5, pp. 311–801, 2014.

[2] A. H. Sayed, “Diffusion adaptation over networks,” in *Array and Statistical Signal Processing, Academic Press Library in Signal Processing*, vol. 3, A. M. Zoubir, M. Viberg, R. Chellappa, and S. Theodoridis, Eds. Cambridge, MA, USA: Academic Press, 2014, pp. 323–453.

[3] R. Nassif, S. Vlaski, C. Richard, J. Chen, and A. H. Sayed, “Multitask learning over graphs: An approach for distributed, streaming machine learning,” *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 14–25, May 2020.

[4] M. H. Cintuglu and D. Ishchenko, “Secure distributed state estimation for networked microgrids,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8046–8055, Oct. 2019.

[5] S. Ghazanfari-Rad and F. Labeau, “Formulation and analysis of LMS adaptive networks for distributed estimation in the presence of transmission errors,” *IEEE Internet Things J.*, vol. 3, no. 2, pp. 146–160, Apr. 2016.

[6] Y. Chen, S. Kar, and J. M. F. Moura, “The Internet of Things: Secure distributed inference,” *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 64–75, Sep. 2018.

[7] N. Bosowski, V. Ingle, and D. Manolakis, “Generalized linear models for count time series,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*. 2017, pp. 4272–4276.

[8] L. Wang and Y. Chi, “Stochastic approximation and memory-limited subspace tracking for Poisson streaming data,” *IEEE Trans. Signal Process.*, vol. 66, no. 4, pp. 1051–1064, Feb. 2018.

[9] P. M. Djurić and C. Richard, *Cooperative and Graph Signal Processing*. Amsterdam, Netherlands: Elsevier, 2018.

[10] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, “Diffusion recursive least-squares for distributed estimation over adaptive networks,” *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[11] S. S. Dias and M. G. Bruno, “Distributed Bernoulli filters for joint detection and tracking in sensor networks,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 3, pp. 260–275, Sep. 2016.

[12] F. S. Cattivelli and A. H. Sayed, “Diffusion LMS strategies for distributed estimation,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[13] J. Plata-Chaves, N. Bogdanović, and K. Berberidis, “Distributed diffusion-based LMS for node-specific adaptive parameter estimation,” *IEEE Trans. Signal Process.*, vol. 63, no. 13, pp. 3448–3460, Jul. 2015.

[14] J. Plata-Chaves, M. H. Bahari, M. Moonen, and A. Bertrand, “Unsupervised diffusion-based LMS for node-specific parameter estimation over wireless sensor networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 4159–4163.

[15] W. Huang, X. Yang, D. Liu, and S. Chen, “Diffusion LMS with component-wise variable step-size over sensor networks,” *IET Signal Process.*, vol. 10, no. 1, pp. 37–45, Feb. 2016.

[16] M. G. Bruno and S. S. Dias, “Collaborative emitter tracking using Rao-Blackwellized random exchange diffusion particle filtering,” *EURASIP J. Adv. Signal Process.*, vol. 2014, no. 1, pp. 1–18, Feb. 2013.

[17] K. Dedecius and P. M. Djurić, “Diffusion filtration with approximate Bayesian computation,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2014, pp. 3207–3211.

[18] W. Li, Z. Wang, Y. Yuan, and L. Guo, “Particle filtering with applications in networked systems: A survey,” *Complex Intell. Syst.*, vol. 2, no. 4, pp. 293–315, Oct. 2016.

[19] F. S. Cattivelli and A. H. Sayed, “Diffusion strategies for distributed Kalman filtering and smoothing,” *IEEE Trans. Automat. Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.

[20] J. Hu, L. Xie, and C. Zhang, “Diffusion Kalman filtering based on covariance intersection,” *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 891–902, Feb. 2012.

[21] R. A. Davis, S. H. Holan, R. Lund, and N. Ravishanker, Eds., *Handbook of Discrete-Valued Time Series*. Boca Raton, FL, USA: Chapman and Hall/CRC, 2016.

[22] C. Weiß, *An Introduction to Discrete-Valued Time Series*. Chichester, U.K.: Wiley, 2018.

[23] D. Manolakis, N. Bosowski, and V. K. Ingle, “Count time-series analysis: A signal processing perspective,” *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 64–81, May 2019.

[24] J. A. Nelder and R. W. M. Wedderburn, “Generalized linear models,” *J. Roy. Stat. Soc. Ser. A (General)*, vol. 135, no. 3, pp. 370–384, 1972.

[25] P. McCullagh and J. A. Nelder, *Generalized Linear Models, Monographs on Statistics and Applied Probability*. London, U.K.: Chapman and Hall, 1989.

[26] R. H. Myers, D. C. Montgomery, G. G. Vining, and T. J. Robinson, *Generalized Linear Models: With Applications in Engineering and the Sciences* (Wiley Series in Probability and Statistics). Hoboken, NJ, USA: Wiley, 2010.

[27] K. Dedecius and R. Žemlička, “Sequential Poisson regression in diffusion networks,” *IEEE Signal Process. Lett.*, vol. 27, pp. 625–629, 2020.

[28] H. K. Lim, W. K. Li, and P. L. H. Yu, “Zero-inflated Poisson regression mixture model,” *Comput. Statist. Data Anal.*, vol. 71, pp. 151–158, 2014.

[29] D. Lambert, “Zero-inflated Poisson regression, with an application to defects in manufacturing,” *Technometrics*, vol. 34, no. 1, pp. 1–14, Feb. 1992.

[30] J. Haslett, A. C. Parnell, J. Hinde, and R. Andrade Moral, “Modelling excess zeros in count data: A new perspective on modelling approaches,” *Int. Stat. Rev.*, vol. 90, pp. 216–236, 2022.

[31] M. Wedel, W. S. Desarbo, J. R. Bult, and V. Ramaswamy, “A latent class Poisson regression model for heterogeneous count data,” *J. Appl. Econometrics*, vol. 8, no. 4, pp. 397–411, 1993.

[32] P. Wang, M. L. Puterman, I. Cockburn, and N. Le, “Mixed poisson regression models with covariance dependent rates,” *Biometrics*, vol. 52, no. 2, pp. 381–400, 1996.

[33] J. N. Gonçalves and W. Barreto-Souza, “Flexible regression models for counts with high-inflation of zeros,” *Metron*, vol. 78, no. 1, pp. 71–95, 2020.

[34] Q. Zhang and A. B. Chan, “Wide-area crowd counting: Multi-view fusion networks for counting in large scenes,” *Int. J. Comput. Vis.*, vol. 130, pp. 1938–1960, 2022.

[35] M. Bernas, B. Płaczek, W. Korski, P. Loska, J. Smyła, and P. Szymała, “A survey and comparison of low-cost sensing technologies for road traffic monitoring,” *Sensors*, vol. 18, no. 10, Sep. 2018, Art. no. 3243.

[36] M. Carminati, O. Kanoun, S. L. Ullo, and S. Marcuccio, “Prospects of distributed wireless sensor networks for urban environmental monitoring,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 6, pp. 44–52, Jun. 2019.

[37] D. M. Titterington, A. F. M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*. Hoboken, NJ, USA: Wiley, 1985.

[38] M. Kárný et al., *Optimized Bayesian Dynamic Advising: Theory and Algorithms*. London, U.K.: Springer, 2006.

[39] R. L. Smith, “Bayesian and frequentist approaches to parametric predictive inference,” in *Bayesian Statistics*, vol. 6, J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, Eds., Oxford, U.K.: Oxford Univ. Press, 1999, pp. 589–612.

[40] M. S. Bartlett and D. Kendall, “The statistical analysis of variance-heterogeneity and the logarithmic transformation,” *J. Roy. Stat. Society. Ser. B. (Stat. Methodol.)*, vol. 8, no. 1, pp. 128–138, 1946.

[41] G. El-Sayyad, “Bayesian and classical analysis of Poisson regression,” *J. Roy. Stat. Society. Ser. B (Stat. Methodol)*, vol. 35, no. 3, pp. 445–451, Jul. 1973.

[42] A. B. Chan and N. Vasconcelos, “Counting people with low-level features and Bayesian regression,” *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2160–2177, Apr. 2012.

[43] S. Haykin, *Adaptive Filter Theory*, 3rd ed. New York, USA: Prentice Hall, 1996.

[44] K. Dedecius, I. Nagy, and M. Kárný, “Parameter tracking with partial forgetting method,” *Int. J. Adaptive Control Signal Process.*, vol. 26, no. 1, pp. 1–12, 2012.

[45] V. Peterka, “Bayesian approach to system identification,” in *Trends and Progress in System Identification*, P. Eykhoff, Ed. Oxford, U.K.: Pergamon Press, 1981, pp. 239–304.

[46] A. W. van der Vaart, *Asymptotic Statistics*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[47] K. Dedecius and P. M. Djurić, “Sequential estimation and diffusion of information over networks: A Bayesian approach with exponential family of distributions,” *IEEE Trans. Signal Process.*, vol. 65, no. 7, pp. 1795–1809, Apr. 2017.

[48] F. Nielsen and R. Nock, “Sided and symmetrized Bregman centroids,” *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2882–2904, Jun. 2009.

[49] J. M. Hilbe, *Modeling Count Data*. New York, NY, USA: Cambridge Univ. Press, 2014.

[50] G. Shmueli, T. P. Minka, J. B. Kadane, S. Borle, and P. Boatwright, “A useful distribution for fitting discrete data: Revival of the Conway-Maxwell-Poisson distribution,” *J. Roy. Stat. Society, Ser. C (Appl. Statist.)*, vol. 54, no. 1, pp. 127–142, Jan. 2005.

[51] S. Ghosal, J. K. Ghosh, and T. Samanta, “On convergence of posterior distributions,” *Ann. Statist.*, vol. 23, no. 6, pp. 2145–2152, 1995.