Learning Methods for Dynamic Topic Modeling in Automated Behavior Analysis

Olga Isupova, Danil Kuzin, and Lyudmila Mihaylova, *Senior Member, IEEE*

***Abstract*— Semisupervised and unsupervised systems provide operators with invaluable support and can tremendously reduce the operators' load. In the light of the necessity to process large volumes of video data and provide autonomous decisions, this paper proposes new learning algorithms for activity analysis in video. The activities and behaviors are described by a dynamic topic model. Two novel learning algorithms based on the expectation maximization approach and variational Bayes inference are proposed. Theoretical derivations of the posterior estimates of model parameters are given. The designed learning algorithms are compared with the Gibbs sampling inference scheme introduced earlier in the literature. A detailed comparison of the learning algorithms is presented on real video data. We also propose an anomaly localization procedure, elegantly embedded in the topic modeling framework. It is shown that the developed learning algorithms can achieve 95% success rate. The proposed framework can be applied to a number of areas, including transportation systems, security, and surveillance.**

***Index Terms*— Behavior analysis, expectation maximization, learning dynamic topic models, unsupervised learning, variational Bayesian approach, video analytics.**

I. Introduction

BEHAVIOR analysis is an important area in intelligent video surveillance, where abnormal behavior detection is a difficult problem. One of the challenges in this field is informality of the problem formulation. Due to the broad scope of applications and desired objectives, there is no unique way, in which normal or abnormal behavior can be described. In general, the objective is to detect unusual events and inform in due course a human operator about them.

This paper considers a probabilistic framework for anomaly detection, where less probable events are labeled as abnormal. We propose two learning algorithms and an anomaly localization procedure for spatial detection of abnormal behaviors.

A. Related Work

There is a wealth of methods for abnormal behavior detection, for example, pattern-based methods [1]–[3]. These methods extract explicit patterns from data and use them as behavior templates for decision-making. In [1], the sum of the visual features of a reference frame is treated as a normal behavior template. Another common approach for representing normal templates is using clusters of visual features [2], [3]. Visual features can range from raw intensity values of pixels to complex features that exploit the data nature [4].

In the testing stage, new observations are compared with the extracted patterns. The comparison is based on some similarity measure between observations, e.g., the Jensen–Shannon divergence in [5] or the Z-score value in [2] and [3]. If the distance between the new observation and any of the normal patterns is larger than a threshold, then the observation is classified as abnormal.

Abnormal behavior detection can be considered as a classification problem. It is difficult in advance to collect and label all kind of abnormalities. Therefore, only one-class label can be expected and one-class classifiers are applied to abnormal behavior detection, e.g., a one-class support vector machine [6], a support vector data description algorithm [7], a neural network approach [8], and a level set method [9] for normal data boundary determination [10].

Another class of methods relies on the estimation of probability distributions of the visual data. These estimated distributions are then used in the decision-making process. Different kinds of probability estimation algorithms are proposed in the literature, e.g., based on nonparametric sample histograms [11], Gaussian distribution modeling [12]. Spatio-temporal motion data dependence is modeled as a coupled Hidden Markov Model (HMM) in [13]. Autoregressive process modeling based on self-organized maps is proposed in [14].

An efficient approach is to seek for feature sets that tend to appear together. These feature sets form typical activities or behaviors in the scene. Topic modeling [15], [16] is an approach to find such kinds of statistical regularities in a form of probability distributions. The approach can be applied for abnormal behavior detection (see [17]–[19]). A number of variations of the conventional topic models for abnormal behavior detection have been recently proposed: clustering of activity distributions [20]; modeling temporal dependencies among activities [21]; and a continuous model for an object velocity [22].

Within the probabilistic modeling approach [12], [13], [17], [18], [20], [22] the decision about abnormality is mainly made by computing likelihood of a new observation. The comparison of the different abnormality measures based on the likelihood estimation is provided in [19].

Topic modeling is originally developed for text mining [15], [16]. It aims to find latent variables called "topics" given the collection of unlabeled text *documents* consisted of *words*. In probabilistic topic modeling documents are represented as a mixture of topics, where each topic is assumed to be a distribution over words.

Manuscript received July 28, 2016; revised March 5, 2017; accepted July 25, 2017. Date of publication September 27, 2017; date of current version August 20, 2018. The work of O. Isupova was supported by the EC Seventh Framework Programme [FP7 2013-2017] TRAcking in compleX sensor systems under Grant 607400. The work of L. Mihaylova was supported in part by the EC Seventh Framework Programme [FP7 2013-2017] TRAcking in compleX sensor systems under Grant 607400 and in part by the U.K. Engineering and Physical Sciences Research Council for the support through the Bayesian Tracking and Reasoning over Time under Grant EP/K021516/1. *(Corresponding author: Olga Isupova.)*

The authors are with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S10 2TN, U.K. (e-mail: o.isupova@sheffield.ac.uk; dkuzin1@sheffield.ac.uk; l.s.mihaylova@sheffield.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2017.2735364

There are two main types of topic models: probabilistic latent semantic analysis (PLSA) [15] and latent Dirichlet allocation (LDA) [16]. The former considers the problem from the frequentist perspective, while the later studies it within the Bayesian approach. The main learning techniques proposed for these models include maximum likelihood estimation via the Expectation–Maximization (EM) algorithm [15], variational Bayes (VB) inference [16], Gibbs sampling (GS) [23], and maximum a posteriori (MAP) estimation [24].

B. Contributions

In this paper, inspired by ideas from [21], we propose an unsupervised learning framework based on a Markov Clustering Topic Model (MCTM) for behavior analysis and anomaly detection. It groups possible topic mixtures of visual documents and forms a Markov chain for the groups.

The key contributions of this paper consist in developing new learning algorithms, namely MAP estimation using the EM algorithm and VB inference for the MCTM, and in proposing an anomaly localization procedure that follows concepts of probabilistic topic modeling. We derive the likelihood expressions as a normality measure of newly observed data. The developed learning algorithms are compared with the GS scheme proposed in [21]. A comprehensive analysis of the algorithms is presented over real video sequences. The empirical results show that the proposed methods provide more accurate results than the GS scheme in terms of anomaly detection performance.

Our preliminary results with the EM algorithm for behavior analysis are published in [25]. In contrast to [25] we now consider a fully Bayesian framework, where we propose the EM algorithm for MAP estimates rather than the maximum likelihood ones. We also propose here a novel learning algorithm based on VB inference and a novel anomaly localization procedure. The experiments are performed on more challenging data sets in comparison to [25].

The rest of this paper is organized as follows. Section II describes the overall structure of visual documents and visual words. Section III introduces the dynamic topic model. The new learning algorithms are presented in Section IV, where the proposed MAP estimation via the EM algorithm and VB algorithm are introduced first and then the GS scheme is reviewed. The methods are given with a detailed discussion about their similarities and differences. The anomaly detection procedure is presented in Section V. The learning algorithms are evaluated with real data in Section VI, and Section VII concludes this paper.

II. Video Analytics Within the Topic Modeling Framework

Video analytics tasks can be formulated within the framework of topics modeling. This requires a definition of visual documents and visual words (see [20], [21]). The whole video sequence is divided into nonoverlapping short clips. These clips are treated as visual documents. Each frame is divided next into grid cells of pixels. Motion detection is applied to each of the cells. The cells where motion is detected are called moving cells. For each of the moving cells the motion direction is determined. This direction is further quantized into four dominant ones—up, left, down, and right (see Fig. 1). The position of the moving cell and the quantized direction of its motion form a visual word.

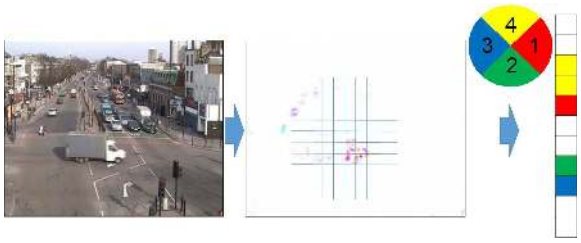


Fig. 1. Structure of the visual feature extraction. From an input frame (left), a map of local motions is calculated (center). The motion is quantized into four directions to get the feature representation (right).

Each of the visual documents is then represented as a sequence of visual words' identifiers, where identifiers are obtained by some ordering of a set of unique words. This discrete representation of the input data can be processed by topic modeling methods.

III. Markov Clustering Topic Model for Behavioral Analysis

A. Motivation

In topic modeling, there are two main kinds of distributions—the distributions over words, which correspond to topics, and the distributions over topics, which characterize the documents. The relationship between documents and words is then represented via latent low-dimensional entities called topics. Having only an unlabeled collection of documents, topic modeling methods restore a hidden structure of data, i.e., the distributions over words and the distributions over topics.

Consider a set of distributions over topics and a topic distribution for each document is chosen from this set. If the cardinality of the set of distributions over topics is less than the number of documents, then documents are clustered into groups such that documents have the same topic distribution within a group. A unique distribution over topics is called a *behavior* in this paper. Therefore, each document corresponds to one behavior. In topic modeling, a document is fully described by a corresponding distribution over topics, which means in this case a document is fully described by a corresponding behavior.

There are a number of applications where we can observe documents clustered into groups with the same distribution over topics. Let us consider some examples from video analytics where a visual word corresponds to a motion within a tiny cell. As topics represent words that statistically often appear together, in video analytics applications topics define some motion patterns in local areas.

Let us consider a road junction regulated by traffic lights. A general motion on the junction is the same with the same traffic light regime. Therefore, the documents associated with the same traffic light regimes have the same distributions over topics, i.e., they correspond to the same behaviors.

Another example is a video stream generated by a video surveillance camera from a train station. Here it is also possible to distinguish several types of general motion within the camera scene: getting off and on a train and waiting for it. These types of motion correspond to behaviors, where the different visual documents showing different instances of the same behavior have very similar motion structures, i.e., the same topic distribution.

Each action in real life lasts for some time, e.g., a traffic light regime stays the same and people get on and off a train for several seconds. Moreover, often these different types of motion or behaviors follow a cycle and their changes occur in some order. These insights motivate to model a sequence of behaviors as a Markov chain, so that the behaviors remain the same during some documents and change in a predefined order. The model that has these described properties is called an MCTM in [21]. The next section formally formulates the model.

B. Model Formulation

This section starts from the introduction of the main notations used through this paper. Denote by $\mathcal{X}$ the vocabulary of all visual words, by $\mathcal{Y}$ the set of all topics, by $\mathcal{Z}$ the set of all behaviors, and x, y, and z are used for elements from these sets, respectively. When an additional element of a set is required, it is denoted with a prime, e.g., z' is another element from $\mathcal{Z}$.

Let $\mathbf{x}_t = \{x_{i,t}\}_{i=1}^{N_t}$ be a set of words for the document t, where N_t is the length of the document t. Let $\mathbf{x}_{1:T_{tr}} = \{\mathbf{x}_t\}_{t=1}^{T_{tr}}$ denote a set of all words for the whole data set, where T_{tr} is the number of documents in the data set. Similarly, denote by $\mathbf{y}_t = \{y_{i,t}\}_{i=1}^{N_t}$ and $\mathbf{y}_{1:T_{tr}} = \{\mathbf{y}_t\}_{t=1}^{T_{tr}}$ a set of topics for the document t and a set of all topics for the whole data set, respectively. Let $\mathbf{z}_{1:T_{tr}} = \{z_t\}_{t=1}^{T_{tr}}$ be a set of all behaviors for all documents.

Note that x, y, and z without subscript denote possible values for a word, topic, and behavior from $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, respectively, while the symbols with subscript denote word, topic, and behavior assignments in particular places in a data set.

Here, $\boldsymbol{\Phi}$ is a matrix corresponding to the distributions over words given the topics, $\boldsymbol{\Theta}$ is a matrix corresponding to the distributions over topics given behaviors. For a Markov chain of behaviors, a vector $\boldsymbol{\pi}$ for a behavior distribution for the first document and a matrix $\boldsymbol{\Xi}$ for transition probability distributions between the behaviors are introduced

$$\begin{aligned}
&\boldsymbol{\Phi} = \{\phi_{x,y}\}_{x\in\mathcal{X},y\in\mathcal{Y}}, \quad \phi_{x,y} = p(x|y), \ \boldsymbol{\phi}_y = \{\phi_{x,y}\}_{x\in\mathcal{X}} \\
&\boldsymbol{\Theta} = \{\theta_{y,z}\}_{y\in\mathcal{Y},z\in\mathcal{Z}}, \quad \theta_{y,z} = p(y|z), \ \boldsymbol{\theta}_z = \{\theta_{y,z}\}_{y\in\mathcal{Y}} \\
&\boldsymbol{\pi} = \{\pi_z\}_{z\in\mathcal{Z}}, \quad \pi_z = p(z) \\
&\boldsymbol{\Xi} = \{\xi_{z',z}\}_{z'\in\mathcal{Z},z\in\mathcal{Z}}, \quad \xi_{z',z} = p(z'|z), \ \boldsymbol{\xi}_z = \{\xi_{z',z}\}_{z'\in\mathcal{Z}}
\end{aligned}$$

where the matrices $\boldsymbol{\Phi}$, $\boldsymbol{\Theta}$, and $\boldsymbol{\Xi}$ and the vector $\boldsymbol{\pi}$ are formed as follows. An element of a matrix on the ith row and jth column is a probability of the ith element given the jth one, e.g., $\phi_{x,y}$ is a probability of the word x in the topic y. The columns of the matrices are then distributions for corresponding elements, e.g., $\boldsymbol{\theta}_z$ is a distribution over topics for the behavior z. Elements of the vector $\boldsymbol{\pi}$ are probabilities of behaviors to be chosen by the first document. All these distributions are categorical.

The introduced distributions form a set

$$\boldsymbol{\Omega} = \{\boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\pi}, \boldsymbol{\Xi}\} \tag{1}$$

of model parameters, and they are estimated during a learning procedure.

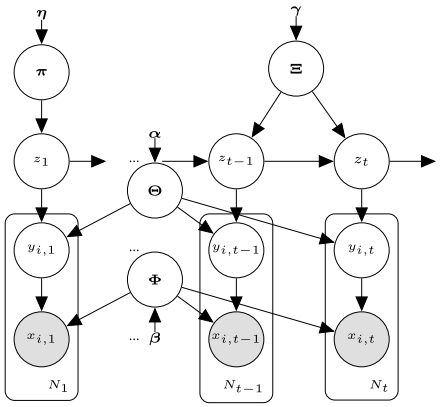


Fig. 2. Graphical representation of the MCTM.

Prior distributions are imposed to all the parameters. Conjugate Dirichlet distributions are used

$$\begin{aligned}
\boldsymbol{\phi}_y &\sim \text{Dir}(\boldsymbol{\phi}_y|\boldsymbol{\beta}), \quad \forall y \in \mathcal{Y} \\
\boldsymbol{\theta}_z &\sim \text{Dir}(\boldsymbol{\theta}_z|\boldsymbol{\alpha}), \quad \forall z \in \mathcal{Z} \\
\boldsymbol{\pi} &\sim \text{Dir}(\boldsymbol{\pi}|\boldsymbol{\eta}) \\
\boldsymbol{\xi}_z &\sim \text{Dir}(\boldsymbol{\xi}_z|\boldsymbol{\gamma}), \quad \forall z \in \mathcal{Z}
\end{aligned}$$

where $\text{Dir}(\cdot)$ is a Dirichlet distribution and $\boldsymbol{\beta}$, $\boldsymbol{\alpha}$, $\boldsymbol{\eta}$, and $\boldsymbol{\gamma}$ are the corresponding hyperparameters. As topics and behaviors are not known a priori and will be specified via the learning procedure, it is impossible to distinguish two topics or two behaviors in advance. This is the reason why all the prior distributions are the same for all topics and all behaviors.

The generative process for the model is as follows. All the parameters are drawn from the corresponding prior Dirichlet distributions. At each time moment t, a behavior z_t is chosen first for a visual document. The behavior is sampled using the matrix $\boldsymbol{\Xi}$ according to the behavior chosen for the previous document. For the first document, the behavior is sampled using the vector $\boldsymbol{\pi}$.

Once the behavior is selected, the procedure of choosing visual words repeats for the number of times equal to the length of the current document N_t. The procedure consists of two steps—sampling a topic $y_{i,t}$ using the matrix $\boldsymbol{\Theta}$ according to the chosen behavior z_t followed by sampling a word $x_{i,t}$ using the matrix $\boldsymbol{\Phi}$ according to the chosen topic $y_{i,t}$ for each token $i \in \{1, \ldots, N_t\}$, where a token is a particular place inside a document where a word is assigned. The generative process is summarized in Algorithm 1. The graphical model, showing the relationships between the variables, can be found in Fig. 2.

The full likelihood of the observed variables $\mathbf{x}_{1:T_{tr}}$, the hidden variables $\mathbf{y}_{1:T_{tr}}$ and $\mathbf{z}_{1:T_{tr}}$, and the set of parameters $\boldsymbol{\Omega}$ can be written then as follows:

$$\begin{aligned}
&p(\mathbf{x}_{1:T_{tr}}, \mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}, \boldsymbol{\Omega}|\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\eta}, \boldsymbol{\gamma}) \\
&= \underbrace{p(\boldsymbol{\pi}|\boldsymbol{\eta})\, p(\boldsymbol{\Xi}|\boldsymbol{\gamma})\, p(\boldsymbol{\Theta}|\boldsymbol{\alpha})\, p(\boldsymbol{\Phi}|\boldsymbol{\beta})}_{\text{Priors}} \\
&\times \underbrace{p(z_1|\boldsymbol{\pi}) \left[\prod_{t=2}^{T_{tr}} p(z_t|z_{t-1}, \boldsymbol{\Xi})\right] \prod_{t=1}^{T_{tr}} \prod_{i=1}^{N_t} p(x_{i,t}|y_{i,t}, \boldsymbol{\Phi}) p(y_{i,t}|z_t, \boldsymbol{\Theta})}_{\text{Likelihood}}.
\end{aligned} \tag{2}$$

In [21], GS is implemented for parameters learning in the MCTM. We propose two new learning algorithms: based on

Algorithm 1 Generative Process for the MCTM

Require: The number of clips – T_{tr}, the length of each clip – N_t $\forall t = \{1, \ldots, T_{tr}\}$, the hyperparameters – $\boldsymbol{\beta}$, $\boldsymbol{\alpha}$, $\boldsymbol{\eta}$, $\boldsymbol{\gamma}$;
Ensure: The data set $\mathbf{x}_{1:T_{tr}} = \{x_{1,1}, \ldots, x_{i,t}, \ldots, x_{N_{T_{tr}},T_{tr}}\}$;
for all $y \in \mathcal{Y}$ **do**
  draw a word distribution for the topic y:
$$\boldsymbol{\phi}_y \sim Dir(\boldsymbol{\phi}_y|\boldsymbol{\beta});$$
for all $z \in \mathcal{Z}$ **do**
  draw a topic distribution for behavior z:
$$\boldsymbol{\theta}_z \sim Dir(\boldsymbol{\theta}_z|\boldsymbol{\alpha});$$
  draw a transition distribution for behavior z:
$$\boldsymbol{\xi}_z \sim Dir(\boldsymbol{\xi}_z|\boldsymbol{\gamma});$$
draw a behavior probability distribution for the initial document
$$\boldsymbol{\pi} \sim Dir(\boldsymbol{\phi}|\boldsymbol{\eta});$$
for all $t \in \{1, \ldots, T_{tr}\}$ **do**
  if $t = 1$ **then**
    draw a behavior for the document from the initial distribution: $z_t \sim Cat(z_t|\boldsymbol{\pi})$[1];
  else
    draw a behavior for the document based on the behavior of the previous document: $z_t \sim Cat(z_t|\boldsymbol{\xi}_{z_{t-1}})$;
  for all $i \in \{1, \ldots, N_t\}$ **do**
    draw a topic for the token i based on the chosen behavior: $y_{i,t} \sim Cat(y_{i,t}|\boldsymbol{\theta}_{z_t})$;
    draw a visual word for the token i based on the chosen topic: $x_{i,t} \sim Cat(x_{i,t}|\boldsymbol{\phi}_{y_{i,t}})$;

an EM algorithm for the MAP estimates of the parameters and based on VB inference to estimate posterior distributions of the parameters. We introduce the proposed learning algorithms below and briefly reviewed the GS scheme.

IV. Parameters Learning

A. Learning: EM Algorithm Scheme

We propose a learning algorithm for MAP estimates of the parameters based on the EM algorithm [26]. The algorithm consists of repeating E- and M-steps. Conventionally, the EM algorithm is applied to get maximum likelihood estimates. In that case, the M-step is

$$\mathcal{Q}(\boldsymbol{\Omega}, \boldsymbol{\Omega}^{\text{old}}) \longrightarrow \max_{\boldsymbol{\Omega}} \tag{3}$$

where $\boldsymbol{\Omega}^{\text{old}}$ denotes the set of parameters obtained at the previous iteration, and $\mathcal{Q}(\boldsymbol{\Omega}, \boldsymbol{\Omega}^{\text{old}})$ is the expected logarithm of the full likelihood function of the observed and hidden variables

$$\mathcal{Q}(\boldsymbol{\Omega}, \boldsymbol{\Omega}^{\text{old}}) = \mathbb{E}_{p(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}})} \log p(\mathbf{x}_{1:T_{tr}}, \mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}|\boldsymbol{\Omega}). \tag{4}$$

[1]Here, $Cat(\cdot|\mathbf{v})$ denotes a categorical distribution, where components of a vector $\mathbf{v}$ are probabilities of a discrete random variable to take one of possible values.

The subscript of the expectation sign means the distribution, with respect to which the expectation is calculated. During the E-step, the posterior distribution of the hidden variables is estimated given the current estimates of the parameters.

In this paper, the EM algorithm is applied to get MAP estimates instead of traditional maximum likelihood ones. The M-step is modified in this case as

$$\mathcal{Q}(\boldsymbol{\Omega}, \boldsymbol{\Omega}^{\text{old}}) + \log p(\boldsymbol{\Omega}|\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\eta}, \boldsymbol{\gamma}) \longrightarrow \max_{\boldsymbol{\Omega}} \tag{5}$$

where $p(\boldsymbol{\Omega}|\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\eta}, \boldsymbol{\gamma})$ is the prior distribution of the parameters.

As the hidden variables are discrete, the expectation converts to a sum of all possible values for the whole set of the hidden variables $\{\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}\}$. The substitution of the likelihood expression from (2) into (5) allows to marginalize some hidden variables from the sum. The remaining distributions that are required for computing the $\mathcal{Q}$-function are as follows:

- $p(z_1 = z|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}})$—the posterior distribution of a behavior for the first document;
- $p(z_t = z', z_{t-1} = z|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}})$—the posterior distribution of two behaviors for successive documents;
- $p(y_{i,t} = y|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}})$—the posterior distribution of a topic assignment for a given token;
- $p(y_{i,t} = y, z_t = z|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}})$—the joint posterior distribution of a topic and behavior assignments for a given token.

With the fixed current values for these posterior distributions the estimates of the parameters that maximize the required functional of the M-step (5) can be computed as

$$\widehat{\phi}^{\text{EM}}_{x,y} = \frac{\left(\beta_x + \hat{n}^{\text{EM}}_{x,y} - 1\right)_+}{\sum_{x' \in \mathcal{X}} \left(\beta_{x'} + \hat{n}^{\text{EM}}_{x',y} - 1\right)_+}, \quad \forall x \in \mathcal{X}, \ y \in \mathcal{Y} \tag{6}$$

$$\widehat{\theta}^{\text{EM}}_{y,z} = \frac{\left(\alpha_y + \hat{n}^{\text{EM}}_{y,z} - 1\right)_+}{\sum_{y' \in \mathcal{Y}} \left(\alpha_{y'} + \hat{n}^{\text{EM}}_{y',z} - 1\right)_+}, \quad \forall y \in \mathcal{Y}, \ z \in \mathcal{Z} \tag{7}$$

$$\widehat{\xi}^{\text{EM}}_{z',z} = \frac{\left(\gamma_{z'} + \hat{n}^{\text{EM}}_{z',z} - 1\right)_+}{\sum_{\check{z} \in \mathcal{Z}} \left(\gamma_{\check{z}} + \hat{n}^{\text{EM}}_{\check{z},z} - 1\right)_+}, \quad \forall z', \ z \in \mathcal{Z} \tag{8}$$

$$\widehat{\pi}^{\text{EM}}_{z} = \frac{\left(\eta_z + \hat{n}^{\text{EM}}_{z} - 1\right)_+}{\sum_{z' \in \mathcal{Z}} \left(\eta_{z'} + \hat{n}^{\text{EM}}_{z'} - 1\right)_+}, \quad \forall z \in \mathcal{Z} \tag{9}$$

where $(a)_+ \stackrel{\text{def}}{=} \max(a, 0)$ [27]; β_x, α_y, and $\gamma_{z'}$ are the elements of the hyperparameter vectors $\boldsymbol{\beta}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\gamma}$, respectively; $\hat{n}^{\text{EM}}_{x,y} = \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} p(y_{i,t} = y|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}}) \mathbb{I}(x_{i,t} = x)$ is the expected number of times, when the word x is associated with the topic y, where $\mathbb{I}(\cdot)$ is the indicator function; $\hat{n}^{\text{EM}}_{y,z} = \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} p(y_{i,t} = y, z_t = z|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}})$ is the expected number of times, when the topic y is associated with the behavior z; $\hat{n}^{\text{EM}}_{z} = p(z_1 = z|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}})$ is the "expected number of times," when the behavior z is associated with the first document, in this case the "expected number" is just a probability, the notation is used for the similarity with the rest of the parameters; and $\hat{n}^{\text{EM}}_{z',z} = \sum_{t=2}^{T_{tr}} p(z_t = z', z_{t-1} = z|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}})$ is the expected number of times, when the behavior z is followed by the behavior z'.

During the E-step with the fixed current estimates of the parameters $\boldsymbol{\Omega}^{\text{old}}$, the updated values for the posterior distributions of the hidden variables should be computed. The

derivation of the updated formulas for these distributions is similar to the Baum–Welch forward–backward algorithm [28], where the EM algorithm is applied to the maximum likelihood estimates for a HMM. This similarity appears because the generative model can be viewed as extension of a HMM.

For effective computation of the required posterior distributions, the additional variables $\acute{\alpha}_z(t)$ and $\acute{\beta}_z(t)$ are introduced. A dynamic programming technique is applied for computation of these variables. Having the updated values for $\acute{\alpha}_z(t)$ and $\acute{\beta}_z(t)$, one can update the required posterior distributions of the hidden variables. The E-step is then formulated as follows (for simplification of notation the superscript "old" for the parameters variables is omitted inside the formulas):

$$\begin{cases} \acute{\alpha}_z(t) = \prod_{i=1}^{N_t} \sum_{y\in\mathcal{Y}} \phi_{x_{i,t},y}\,\theta_{y,z} \\ \qquad \times \sum_{z'\in\mathcal{Z}} \acute{\alpha}_{z'}(t-1)\xi_{z,\tilde{z}}, \quad \text{if } t \geq 2 \\ \acute{\alpha}_z(1) = \pi_z \prod_{i=1}^{N_1} \sum_{y\in\mathcal{Y}} \phi_{x_{i,1},y}\,\theta_{y,z} \end{cases} \tag{10}$$

$$\begin{cases} \acute{\beta}_z(t) = \sum_{z'\in\mathcal{Z}} \acute{\beta}_{z'}(t+1)\xi_{z',z} \\ \qquad \times \prod_{i=1}^{N_{t+1}} \sum_{y\in\mathcal{Y}} \phi_{x_{i,t+1},y}\,\theta_{y,z'}, \quad \text{if } t < T_{tr} \\ \acute{\beta}_z(T_{tr}) = 1 \end{cases} \tag{11}$$

$$K = \sum_{z\in\mathcal{Z}} \acute{\alpha}_z(1)\acute{\beta}_z(1) \tag{12}$$

$$p(z_1|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}}) = \frac{\acute{\alpha}_{z_1}(1)\acute{\beta}_{z_1}(1)}{K} \tag{13}$$

$$p(z_t, z_{t-1}|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}}) = \frac{\acute{\alpha}_{z_{t-1}}(t-1)\acute{\beta}_{z_t}(t)\xi_{z_t,z_{t-1}}}{K} \times \prod_{i=1}^{N_t} \sum_{y\in\mathcal{Y}} \phi_{x_{i,t},y}\theta_{y,z_t} \tag{14}$$

$$\begin{cases} p(y_{i,t}, z_t|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}}) = \dfrac{\phi_{x_{i,t},y_{i,t}}\theta_{y_{i,t},z_t}\acute{\beta}_{z_t}(t)}{K} \\ \quad \times \sum_{z'\in\mathcal{Z}} \acute{\alpha}_{z'}(t-1)\xi_{z_t,z'} \prod_{\substack{j=1 \\ j\neq i}}^{N_t} \sum_{y'\in\mathcal{Y}} \phi_{x_{j,t},y'}\theta_{y',z_t}, \quad \text{if } t \geq 2 \\ p(y_{i,1}, z_1|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}}) = \dfrac{\phi_{x_{i,1},y_{i,1}}\theta_{y_{i,1},z_1}\acute{\beta}_{z_1}(1)}{K} \\ \qquad \times \pi_{z_1} \prod_{\substack{j=1 \\ j\neq i}}^{N_1} \sum_{y'\in\mathcal{Y}} \phi_{x_{j,1},y'}\theta_{y',z_1} \end{cases} \tag{15}$$

$$p(y_{i,t}|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}}) = \sum_{z\in\mathcal{Z}} p(y_{i,t}, z|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{old}}) \tag{16}$$

where K is a normalization constant for all the posterior distributions of the hidden variables.

Starting with some random initialization of the parameter estimates, the EM algorithm iterates the E- and M-steps until convergence. The obtained estimates of the parameters are used for further analysis.

B. Learning: Variational Bayes Scheme

We also propose a learning algorithm based on the VB approach [29] to find approximated posterior distributions for both the hidden variables and the parameters.

In the VB inference scheme, the true posterior distribution, in this case the distribution of the parameters and the hidden variables $p(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}, \boldsymbol{\Omega}|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, is approximated with a factorized distribution—$q(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}, \boldsymbol{\Omega})$. The approximation is made to minimize the Kullback–Leibler divergence between the factorized distribution and true one. We factorize the distribution in order to separate the hidden variables and the parameters

$$\begin{aligned} &\hat{q}(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}, \boldsymbol{\Omega}) \\ &\quad = \hat{q}(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}})\hat{q}(\boldsymbol{\Omega}) \\ &\quad \stackrel{\text{def}}{=} \operatorname{argmin} \text{KL}(q(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}})q(\boldsymbol{\Omega})|| \\ &\qquad\qquad p(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}, \boldsymbol{\Omega}|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\alpha}, \boldsymbol{\beta})) \end{aligned} \tag{17}$$

where KL denotes the Kullback–Leibler divergence. The minimization of the Kullback–Leibler divergence is equivalent to the maximization of the evidence lower bound. The maximization is done by coordinate ascent [29].

During the update of the parameters, the approximated distribution $q(\boldsymbol{\Omega})$ is further factorized

$$q(\boldsymbol{\Omega}) = q(\boldsymbol{\pi})q(\boldsymbol{\Xi})q(\boldsymbol{\Theta})q(\boldsymbol{\Phi}). \tag{18}$$

Note that this factorization of approximated parameter distributions is a corollary of our model and not an assumption.

The iterative process of updating the approximated distributions of the parameters and the hidden variables can be formulated as an EM-like algorithm, where during the E-step, the approximated distributions of the hidden variables are updated and during the M-step, the approximated distributions of the parameters are updated.

The M-like step is as follows:

$$\begin{cases} q(\boldsymbol{\Phi}) = \prod_{y\in\mathcal{Y}} Dir(\boldsymbol{\phi}_y; \tilde{\boldsymbol{\beta}}_y) \\ \tilde{\beta}_{x,y} = \beta_x + \hat{n}^{\text{VB}}_{x,y}, \qquad \forall x \in \mathcal{X}, y \in \mathcal{Y} \end{cases} \tag{19}$$

$$\begin{cases} q(\boldsymbol{\Theta}) = \prod_{z\in\mathcal{Z}} Dir(\boldsymbol{\theta}_z; \tilde{\boldsymbol{\alpha}}_z) \\ \tilde{\alpha}_{y,z} = \alpha_y + \hat{n}^{\text{VB}}_{y,z}, \qquad \forall y \in \mathcal{Y}, z \in \mathcal{Z} \end{cases} \tag{20}$$

$$\begin{cases} q(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi}; \tilde{\boldsymbol{\eta}}) \\ \tilde{\eta}_z = \eta_z + \hat{n}^{\text{VB}}_z, \quad \forall z \in \mathcal{Z} \end{cases} \tag{21}$$

$$\begin{cases} q(\boldsymbol{\Xi}) = \prod_{z\in\mathcal{Z}} Dir(\boldsymbol{\xi}_z; \tilde{\boldsymbol{\gamma}}_z) \\ \tilde{\gamma}_{z',z} = \gamma_{z'} + \hat{n}^{\text{VB}}_{z',z}, \qquad \forall z', z \in \mathcal{Z} \end{cases} \tag{22}$$

where $\tilde{\boldsymbol{\beta}}_y$, $\tilde{\boldsymbol{\alpha}}_z$, $\tilde{\boldsymbol{\eta}}$, and $\tilde{\boldsymbol{\gamma}}_z$ are updated hyperparameters of the corresponding posterior Dirichlet distributions, and $\hat{n}^{\text{VB}}_{x,y} = \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} \mathbb{I}(x_{i,t} = x) q(y_{i,t} = y)$ is the expected number of times, when the word x is associated with the topic y. Here and below the expected number is computed with respect to the approximated posterior distributions of the hidden variables; $\hat{n}^{\text{VB}}_{y,z} = \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} q(y_{i,t} = y, z_t = z)$ is the expected number of times, when the topic y is associated with the behavior z; $\hat{n}^{\text{VB}}_z = q(z_1 = z)$ is the "expected number" of times, when the behavior z is associated with the first document;

and $\hat{n}^{\mathrm{VB}}_{z',z} = \sum_{t=2}^{T_{tr}} q(z_t = z', z_{t-1} = z)$ is the expected number of times, when the behavior z is followed by the behavior z'.

The following additional variables are introduced for the E-like step:

$$\tilde{\pi}_z = \exp\left(\psi(\tilde{\eta}_z) - \psi\left(\sum_{z'\in\mathcal{Z}} \tilde{\eta}_{z'}\right)\right) \tag{23}$$

$$\tilde{\xi}_{\tilde{z},z} = \exp\left(\psi(\tilde{\gamma}_{\tilde{z},z}) - \psi\left(\sum_{z'\in\mathcal{Z}} \tilde{\gamma}_{z',z}\right)\right) \tag{24}$$

$$\tilde{\phi}_{x,y} = \exp\left(\psi(\tilde{\beta}_{x,y}) - \psi\left(\sum_{x'\in\mathcal{X}} \tilde{\beta}_{x',y}\right)\right) \tag{25}$$

$$\tilde{\theta}_{y,z} = \exp\left(\psi(\tilde{\alpha}_{y,z}) - \psi\left(\sum_{y'\in\mathcal{Y}} \tilde{\alpha}_{y',z}\right)\right) \tag{26}$$

where $\psi(\cdot)$ is the digamma function.

Using these additional notations, the E-like step is formulated the same as the E-step of the EM algorithm, replacing everywhere the estimates of the parameters with the corresponding tilde introduced notation and true posterior distributions of the hidden variables with the corresponding approximated ones in (10)–(16).

The point estimates of the parameters can be obtained by expected values of the posterior approximated distributions. An expected value for a Dirichlet distribution (a posterior distribution for all the parameters) is a normalized vector of hyperparameters. Using the expressions for the hyperparameters from (19)–(22), the final parameters' estimates can be obtained by

$$\widehat{\phi}^{\mathrm{VB}}_{x,y} = \frac{\beta_x + \hat{n}^{\mathrm{VB}}_{x,y}}{\sum_{x'\in\mathcal{X}}\left(\beta_{x'} + \hat{n}^{\mathrm{VB}}_{x',y}\right)}, \quad \forall x \in \mathcal{X}, y \in \mathcal{Y} \tag{27}$$

$$\widehat{\theta}^{\mathrm{VB}}_{y,z} = \frac{\alpha_y + \hat{n}^{\mathrm{VB}}_{y,z}}{\sum_{y'\in\mathcal{Y}}\left(\alpha_{y'} + \hat{n}^{\mathrm{VB}}_{y',z}\right)}, \quad \forall y \in \mathcal{Y}, z \in \mathcal{Z} \tag{28}$$

$$\widehat{\xi}^{\mathrm{VB}}_{z',z} = \frac{\gamma_{z'} + \hat{n}^{\mathrm{VB}}_{z',z}}{\sum_{\tilde{z}\in\mathcal{Z}}\left(\gamma_{\tilde{z}} + \hat{n}^{\mathrm{VB}}_{\tilde{z},z}\right)}, \quad \forall z', z \in \mathcal{Z} \tag{29}$$

$$\widehat{\pi}^{\mathrm{VB}}_{z} = \frac{\eta_z + \hat{n}^{\mathrm{VB}}_{z}}{\sum_{z'\in\mathcal{Z}}\left(\eta_{z'} + \hat{n}^{\mathrm{VB}}_{z'}\right)}, \quad \forall z \in \mathcal{Z}. \tag{30}$$

C. Learning: Gibbs Sampling Algorithm

In [21], the collapsed version of GS is used for parameter learning in the MCTM. The Markov chain is built to sample only the hidden variables $y_{i,t}$ and z_t, while the parameters $\boldsymbol{\Phi}$, $\boldsymbol{\Theta}$, and $\boldsymbol{\Xi}$ are integrated out (note that the distribution for the initial behavior choice $\boldsymbol{\pi}$ is not considered in [21]).

During the burn-in stage, the hidden topic and behavior assignments to each token in the data set are drawn from the conditional distributions given all the remaining variables. Following the Markov Chain Monte Carlo framework, it would draw samples from the posterior distribution $p(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}} | \mathbf{x}_{1:T_{tr}}, \boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\eta}, \boldsymbol{\gamma})$. From the whole sample for $\{\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}\}$, the parameters can be estimated as in [23]

$$\widehat{\phi}^{\mathrm{GS}}_{x,y} = \frac{\hat{n}^{\mathrm{GS}}_{x,y} + \beta_x}{\sum_{x'\in\mathcal{X}}\left(\hat{n}^{\mathrm{GS}}_{x',y} + \beta_{x'}\right)}, \quad \forall x \in \mathcal{X}, y \in \mathcal{Y} \tag{31}$$

$$\widehat{\theta}^{\mathrm{GS}}_{y,z} = \frac{\hat{n}^{\mathrm{GS}}_{y,z} + \alpha_y}{\sum_{y'\in\mathcal{Y}}\left(\hat{n}^{\mathrm{GS}}_{y',z} + \alpha_{y'}\right)}, \quad \forall y \in \mathcal{Y}, z \in \mathcal{Z} \tag{32}$$

$$\widehat{\xi}^{\mathrm{GS}}_{z',z} = \frac{\hat{n}^{\mathrm{GS}}_{z',z} + \gamma_{z'}}{\sum_{\tilde{z}\in\mathcal{Z}}\left(\hat{n}^{\mathrm{GS}}_{\tilde{z},z} + \gamma_{\tilde{z}}\right)}, \quad \forall z', z \in \mathcal{Z} \tag{33}$$

where $\hat{n}^{\mathrm{GS}}_{x,y}$ is the count for the number of times, when the word x is associated with the topic y; $\hat{n}^{\mathrm{GS}}_{y,z}$ is the count for the topic y and the behavior z pair; $\hat{n}^{\mathrm{GS}}_{z',z}$ is the count for the number of times, when the behavior z is followed by the behavior z'.

D. Similarities and Differences of the Learning Algorithms

The point parameter estimates for all three learning algorithms (6)–(9), (27)–(30), and (31)–(33) have a similar form. The EM algorithm estimates differ up to the hyperparameters reassignment—adding one to all the hyperparameters in the VB or GS algorithms ends up with the same final equations for the parameters estimates in the EM algorithm. We explore this in the experimental part. This "-1" term in the EM algorithm formulas (6)–(8) occurs because it uses modes of the posterior distributions, while the point estimates obtained by the VB and GS algorithms are means of the corresponding posterior distributions. For a Dirichlet distribution, which is a posterior distribution for all the parameters, mode and mean expressions differ by this "-1" term.

The main differences of the methods consist in the ways the counts $n_{x,y}$, $n_{y,z}$, and $n_{z',z}$ are estimated. In the GS algorithm, they are calculated by a single sample from the posterior distribution of the hidden variables $p(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}} | \mathbf{x}_{1:T_{tr}}, \boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\gamma})$. In the EM algorithm, the counts are computed as expected numbers of the corresponding events with respect to the posterior distributions of the hidden variables. In the VB algorithm, the counts are computed in the same way as in the EM algorithm up to replacing the true posterior distributions with the approximated ones.

Our observations for the dynamic topic model confirm the comparison results for the vanilla PLSA and LDA models provided in [30].

V. Anomaly Detection

This paper presents on-line anomaly detection with the MCTM in video streams. The decision-making procedure is divided into two stages. At a learning stage, the parameters are estimated using T_{tr} visual documents by one of the learning algorithms, presented in Section IV. After that during a testing stage a decision about abnormality of new upcoming testing documents is made comparing a marginal likelihood of each document with a threshold. The likelihood is computed using the parameters obtained during the learning stage. The threshold is a parameter of the method and can be set empirically, for example, to label 2% of the testing data as abnormal. This paper presents a comparison of the algorithms (Section VI) using the measure independent of threshold value selection.

We also propose an anomaly localization procedure during the testing stage for those visual documents that are labeled as abnormal. This procedure is designed to provide spatial

information about anomalies, while documents labeled as abnormal provide temporal detection. The following sections introduce both the anomaly detection procedure on a document level and the anomaly localization procedure within a video frame.

A. Abnormal Documents Detection

The marginal likelihood of a new visual document $\mathbf{x}_{t+1}$ given all the previous data $\mathbf{x}_{1:t}$ can be used as a normality measure of the document [21]

$$p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}) = \iiint p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}, \boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\Xi}) p(\boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{x}_{1:t}) \mathrm{d}\boldsymbol{\Phi}\mathrm{d}\boldsymbol{\Theta}\mathrm{d}\boldsymbol{\Xi}. \tag{34}$$

If the likelihood value is small it means that the current document cannot be fit to the learnt behaviors and topics, which represent typical motion patterns. Therefore, this is an indication for an abnormal event in this document. The decision about abnormality of a document is then made by comparing the marginal likelihood of the document with the threshold.

In real world applications, it is essential to detect anomalies as soon as possible. Hence an approximation of the integral in (34) is used for efficient computation. The first approximation is based on the assumption that the training data set is representative for parameter learning, which means that the posterior probability of the parameters would not change if there is more observed data

$$p(\boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{x}_{1:t}) \approx p(\boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{x}_{1:Tr}) \quad \forall t \geq T_{tr}. \tag{35}$$

The marginal likelihood can be then approximated as

$$\iiint p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}, \boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\Xi}) p(\boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{x}_{1:t}) \mathrm{d}\boldsymbol{\Phi}\mathrm{d}\boldsymbol{\Theta}\mathrm{d}\boldsymbol{\Xi} \approx \iiint p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}, \boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\Xi}) p(\boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{x}_{1:T_{tr}}) \mathrm{d}\boldsymbol{\Phi}\mathrm{d}\boldsymbol{\Theta}\mathrm{d}\boldsymbol{\Xi}. \tag{36}$$

Depending on the algorithm used for learning the integral in (36) can be further approximated in different ways. We consider two types of approximation.

1) Plug-in Approximation: The point estimates of the parameters can be plug-in in the integral (36) for approximation

$$\begin{aligned}&\iiint p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}, \boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\Xi}) p(\boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{x}_{1:Tr}) \mathrm{d}\boldsymbol{\Phi}\mathrm{d}\boldsymbol{\Theta}\mathrm{d}\boldsymbol{\Xi} \\ &\approx \iiint p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}, \boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\Xi}) \delta_{\hat{\boldsymbol{\Phi}}}(\boldsymbol{\Phi}) \delta_{\hat{\boldsymbol{\Theta}}}(\boldsymbol{\Theta}), \delta_{\hat{\boldsymbol{\Xi}}}(\boldsymbol{\Xi}) \mathrm{d}\boldsymbol{\Phi}\mathrm{d}\boldsymbol{\Theta}\mathrm{d}\boldsymbol{\Xi} \\ &= p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}, \hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Xi}})\end{aligned} \tag{37}$$

where $\delta_a(\cdot)$ is the delta-function with the center in a; $\hat{\boldsymbol{\Phi}}$, $\hat{\boldsymbol{\Theta}}$, $\hat{\boldsymbol{\Xi}}$ are point estimates of the parameters, which can be computed by any of the considered learning algorithms using (6)–(8), (27)–(29), or (31)–(33).

The product and sum rules, the conditional independence equations from the generative model are then applied and the final formula for the plug-in approximation is as follows:

$$\begin{aligned}p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}) &\approx p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}, \hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Xi}}) \\ &= \sum_{z_t} \sum_{z_{t+1}} [p(\mathbf{x}_{t+1}|z_{t+1}, \hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{\Theta}}) p(z_{t+1}|z_t, \hat{\boldsymbol{\Xi}}) \\ &\quad \times p(z_t|\mathbf{x}_{1:t}, \hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Xi}})]\end{aligned} \tag{38}$$

where the predictive probability of the behavior for the current document, given the observed data up to the current document, can be computed via the recursive formula

$$p(z_t|\mathbf{x}_{1:t}, \hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Xi}}) = \sum_{z_{t-1}} \frac{p(\mathbf{x}_t|z_t, \hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{\Theta}}) p(z_t|z_{t-1}, \hat{\boldsymbol{\Xi}}) p(z_{t-1}|\mathbf{x}_{1:t-1}, \hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Xi}})}{p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Xi}})}. \tag{39}$$

The point estimates can be computed for all three learning algorithms; therefore, a normality measure based on the plug-in approximation of the marginal likelihood is applicable for all of them.

2) Monte Carlo Approximation: If samples $\{\boldsymbol{\Phi}^s, \boldsymbol{\Theta}^s, \boldsymbol{\Xi}^s\}$ from the posterior distribution $p(\boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{x}_{1:T_{tr}})$ of the parameters can be obtained, the integral (36) is further approximated by the Monte Carlo method

$$\iiint p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}, \boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\Xi}) p(\boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{x}_{1:T_{tr}}) \mathrm{d}\boldsymbol{\Phi}\mathrm{d}\boldsymbol{\Theta}\mathrm{d}\boldsymbol{\Xi} \approx \frac{1}{S} \sum_{s=1}^{S} p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}, \boldsymbol{\Phi}^s, \boldsymbol{\Theta}^s, \boldsymbol{\Xi}^s) \tag{40}$$

where S is the number of samples. These samples can be obtained: 1) from the approximated posterior distributions $q(\boldsymbol{\Phi})$, $q(\boldsymbol{\Theta})$, and $q(\boldsymbol{\Xi})$ of the parameters, computed by the VB learning algorithm or 2) from the independent samples of the GS scheme. For the conditional likelihood $p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}, \boldsymbol{\Phi}^s, \boldsymbol{\Theta}^s, \boldsymbol{\Xi}^s)$, the formula (38) is valid.

Note that for the approximated posterior distribution of the parameters, i.e., the output of the VB learning algorithm, the integral (36) can be resolved analytically, but it would be computationally infeasible. This is the reason why the Monte Carlo approximation is used in this case.

Finally, in order to compare documents of different lengths the normalized likelihood is used as a normality measure s

$$s(\mathbf{x}_{t+1}) = \frac{1}{N_{t+1}} p(\mathbf{x}_{t+1}|\mathbf{x}_{1:t}). \tag{41}$$

B. Localization of Anomalies

The topic modeling approach allows to compute a likelihood function not only of the whole document but of an individual word within the document too. Recall that the visual word contains the information about a location in the frame. We propose to use the location information from the least probable words (e.g., 10 words with the least likelihood values) to localize anomalies in the frame. Note, we do not require anything additional to a topic model, e.g., modeling regional information explicitly as in [31] or comparing a test document with training ones as in [32]. Instead, the proposed anomaly localization procedure is general and can be applied in any topic modeling-based method, where spatial information is encoded to visual words.

The marginal likelihood of a word can be computed in a similar way to the likelihood of the whole document. For the point estimates of the parameters and plug-in approximation of the integral, it is

$$p(x_{i,t+1}|\mathbf{x}_{1:t}) \approx p(x_{i,t+1}|\mathbf{x}_{1:t}, \hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Xi}}). \tag{42}$$

(a) (b)

(c) (d)

Fig. 3. Sample frames of the real data sets. (a) and (b) Two sample frames from the QMUL data. (c) and (d) Two sample frames from the Idiap data.

For the samples from the posterior distributions of the parameters and the Monte Carlo integral approximation, it is

$$p(x_{i,t+1}|\mathbf{x}_{1:t}) \approx \frac{1}{S}\sum_{s=1}^{S} p(x_{i,t+1}|\mathbf{x}_{1:t}, \mathbf{\Phi}^s, \mathbf{\Theta}^s, \mathbf{\Xi}^s). \quad (43)$$

VI. Performance Validation

We compare the two proposed learning algorithms, based on EM and VB, with the GS algorithm, proposed in [21], on two real data sets.

A. Setup

The performance of the algorithms is compared on the QMUL street intersection data [21] and Idiap traffic junction data [19]. Both data sets are 45-min video sequences, captured busy traffic road junctions, where we use a 5-min video sequence as a training data set and others as a testing one. The documents that have less than 20 visual words are discarded from consideration. In practice, these documents can be classified to be normal by default as there is no enough information to make a decision. The frame size for both data sets is 288×360. Sample frames are presented in Fig. 3.

The size of grid cells is set to 8×8 pixels for spatial quantization of the local motion for visual word determination. Nonoverlapping clips with a 1-s length are treated as visual documents.

We also study the influence of the hyperparameters on the learning algorithms. In all the experiments, we use the symmetric hyperparameters: $\boldsymbol{\alpha} = \{\alpha, \ldots, \alpha\}$; $\boldsymbol{\beta} = \{\beta, \ldots, \beta\}$; $\boldsymbol{\gamma} = \{\gamma, \ldots, \gamma\}$; and $\boldsymbol{\eta} = \{\eta, \ldots, \eta\}$. The three groups of the hyperparameters settings are compared: $\{\alpha = 1, \beta = 1, \gamma = 1, \eta = 1\}$ (referred as "prior type 1"); $\{\alpha = 8, \beta = 0.05, \gamma = 1, \eta = 1\}$ ("prior type H"); and $\{\alpha = 9, \beta = 1.05, \gamma = 2, \eta = 2\}$ ("prior type H + 1"). Note that the first group corresponds to the case when in the EM algorithm learning scheme the prior components are canceled out, i.e., the MAP estimates in this case are equal to the maximum likelihood ones. The equations for the point estimates in the EM learning algorithm with the prior type H + 1 of the hyperparameters' settings are equal to the equations for the point estimates in the VB and GS learning algorithms with the prior type H of the settings. The corresponding Dirichlet distributions with all used parameters are presented in Fig. 4.

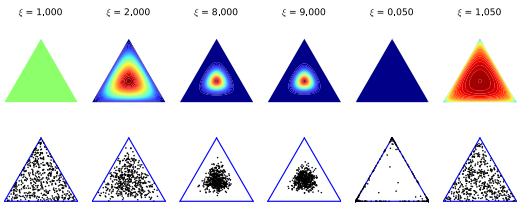


Fig. 4. Dirichlet distributions with different symmetric parameters ξ. For the representation purposes, the 3-D space is used. Colors correspond to the Dirichlet probability density function values in the area (top row). Samples generated from the corresponding density functions (bottom row). The sample size is 500.

Note that parameter learning is an ill-posed problem in topic modeling [27]. This means there is no unique solution for parameter estimates. We use 20 Monte Carlo runs for all the learning algorithms with different random initializations resulting with different solutions. The mean results among these runs are presented below for comparison.

All three algorithms are run with three different groups of hyperparameters' settings. The number of topics and behaviors is set to 8 and 4, respectively, for the QMUL data set, 10 and 3 are used for the corresponding values for the Idiap data set. The EM and VB algorithms are run for 100 iterations. The GS algorithm is run for 500 burn-in iterations and independent samples are taken with a 100-iterations delay after the burn-in period.

B. Performance Measure

Anomaly detection performance of the algorithms depends on threshold selection. To make a fair comparison of the different learning algorithms, we use a performance measure, which is independent of threshold selection.

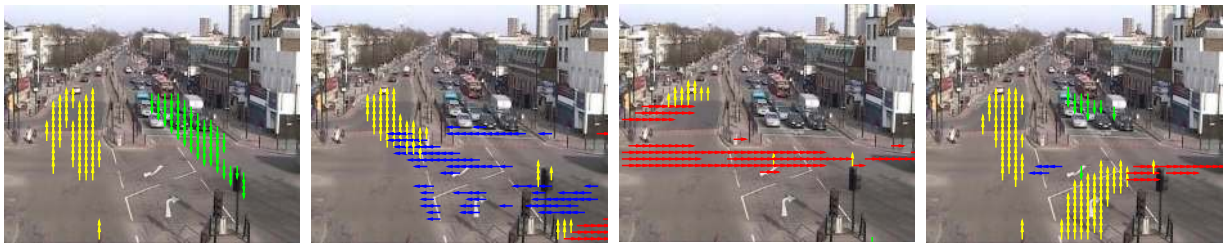
In binary classification, the following measures [28] are used: TP—true positive, a number of documents, which are correctly detected as positive (abnormal in our case); TN—true negative, a number of documents, which are correctly detected as negative (normal in our case); FP—false positive, a number of documents, which are incorrectly detected as positive, when they are negative; FN—false negative, a number of documents, which are incorrectly detected as negative, when they are positive; precision $= (\mathrm{TP}/(\mathrm{TP} + \mathrm{FP}))$—a fraction of correct detections among all documents labeled as abnormal by an algorithm; recall $= (\mathrm{TP}/(\mathrm{TP} + \mathrm{FN}))$—a fraction of correct detections among all truly abnormal documents.

The area under the precision-recall curve is used as a performance measure in this paper. This measure is more informative for detection of rare events than the popular area under the receiver operating characteristic curve [28].

C. Parameter Learning

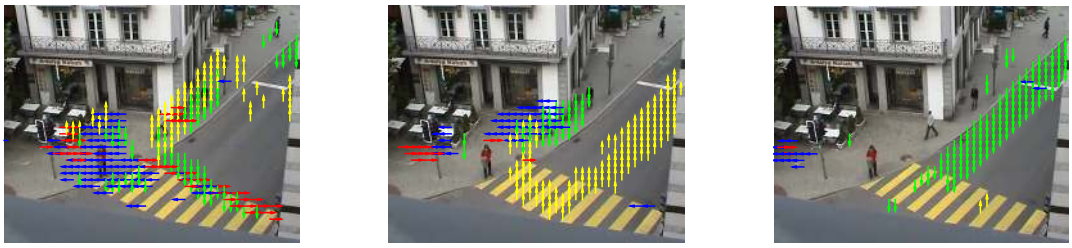
We visualize the learnt behaviors for the qualitative assessment of the proposed framework (Figs. 5 and 6). For illustrative purposes, we consider one run of the EM learning algorithm with the prior type H + 1 of the hyperparameters settings.

The behaviors learnt for the QMUL data are shown in Fig. 5 (for visualization words representing 50% of probability mass



(a) (b) (c) (d)

Fig. 5. Behaviors learnt by the EM learning algorithm for the QMUL data. The arrows represent the visual words: the location and direction of the motion. (a) First behavior corresponds to the vertical traffic flow. (b) Second and (c) third behaviors correspond to the left and the right traffic flow, respectively. (d) Fourth behavior corresponds to turns that follow the vertical traffic flow.



(a) (b) (c)

Fig. 6. Behaviors learnt by the EM learning algorithm for the Idiap data. The arrows represent the visual words: the location and direction of the motion. (a) First behavior corresponds to the pedestrian motion. (b) Second and (c) third behaviors correspond to the upward and downward traffic flows, respectively.

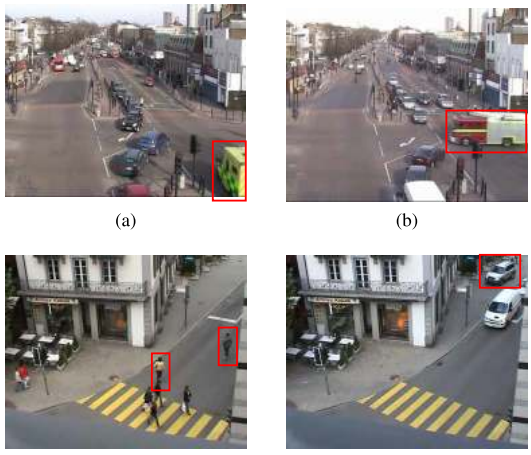
of a behavior are used). One can notice that the algorithm correctly recognizes the motion patterns in the data. The general motion of the scene follows a cycle: a vertical traffic flow [the first behavior in Fig. 5(a)], when cars move downward and upward on the road; left and right turns [the fourth behavior in Fig. 5(d)]: some cars moving on the "vertical" road turn to the perpendicular road at the end of the vertical traffic flow; a left traffic flow [the second behavior in Fig. 5(b)], when cars move from right to left on the "horizontal" road; and a right traffic flow [the third behavior in Fig. 5(c)], when cars move from left to right on the "horizontal" road. Note that the ordering numbers of behaviors correspond to their internal representation in the algorithm. The transition probability matrix $\boldsymbol{\Xi}$ is used to recognize the correct behaviors order in the data.

Fig. 6 presents the behaviors learnt for the Idiap data. In this case, the learnt behaviors have also a clear semantic meaning. The scene motion follows a cycle: a pedestrian flow [the first behavior in Fig. 6(a)], when cars stop in front of the stop line and pedestrians cross the road; a downward traffic flow [the third behavior in Fig. 6(c)], when cars move downward along the road; and an upward traffic flow [the second behavior in Fig. 6(b)], when cars from left and right sides move upward on the road.

D. Anomaly Detection

In this section, the anomaly detection performance achieved by all three learning algorithms is compared. The data sets contain the number of abnormal events, such as jaywalking, car moving on the opposite lane, and disruption of the traffic flow (see Fig. 7).

For the EM learning algorithm, the plug-in approximation of the marginal likelihood is used for anomaly detection. For both the VB and GS learning algorithms, the plug-in and Monte Carlo approximations of the likelihood are used. Note that for the GS algorithm, samples are obtained during the learning stage, the more samples are used for integral approximation the more computational cost of the learning stage. We test 5 and 100 independent samples. For the VB learning algorithm, samples are obtained after the learning stage from the posterior distributions, parameters of which are learnt. This means that the number of samples that are used for anomaly detection does not influence on the computational cost of learning. We test the Monte Carlo approximation of the marginal likelihood with 5 and 100 samples for the VB learning algorithm.



(a) (b) (c) (d)

Fig. 7. Examples of abnormal events. (a) Car moving on the opposite lane. (b) Disruption of the traffic flow. (c) Jaywalking. (d) Car moving on the sidewalk.

As a result, we have 21 methods to compare: obtained by three learning algorithms; three different groups of hyperparameters' settings; one type of marginal likelihood approximation for the EM learning algorithm; and two types of marginal likelihood approximation for the VB and GS learning algorithms, where two Monte Carlo approximations are used with 5 and 100 samples. The list of methods' references can be found in Table I.

Note that we achieve a very fast decision-making performance in our framework. Indeed, anomaly detection

TABLE I
METHODS' REFERENCES

Reference	Learning algorithm	Hyperparameters settings	Marginal likelihood approximation	Number of posterior samples
EM 1 p	EM	type 1	Plug-in	—
EM H p	EM	type H	Plug-in	—
EM H+1 p	EM	type H+1	Plug-in	—
VB 1 p	VB	type 1	Plug-in	—
VB 1 mc 5	VB	type 1	Monte Carlo	5
VB 1 mc 100	VB	type 1	Monte Carlo	100
VB H p	VB	type H	Plug-in	—
VB H mc 5	VB	type H	Monte Carlo	5
VB H mc 100	VB	type H	Monte Carlo	100
VB H+1 p	VB	type H+1	Plug-in	—
VB H+1 mc 5	VB	type H+1	Monte Carlo	5
VB H+1 mc 100	VB	type H+1	Monte Carlo	100
GS 1 p	GS	type 1	Plug-in	—
GS 1 mc 5	GS	type 1	Monte Carlo	5
GS 1 mc 100	GS	type 1	Monte Carlo	100
GS H p	GS	type H	Plug-in	—
GS H mc 5	GS	type H	Monte Carlo	5
GS H mc 100	GS	type H	Monte Carlo	100
GS H+1 p	GS	type H+1	Plug-in	—
GS H+1 mc 5	GS	type H+1	Monte Carlo	5
GS H+1 mc 100	GS	type H+1	Monte Carlo	100

is made for approximately 0.0044 s/visual document by the plug-in approximation of the marginal likelihood, for 0.0177 s/document by the Monte Carlo approximation with 5 samples and for 0.3331 s/document by the Monte Carlo approximation with 100 samples.[2]

The mean areas under precision-recall curves for anomaly detection for all 21 compared methods can be found in Fig. 8. Below we examine the results with respect to hyperparameters sensitivity, an influence of the likelihood approximation on the final performance, we also compare the learning algorithms and discuss anomaly localization results.

1) Hyperparameters Sensitivity: This section presents sensitivity analysis of the anomaly detection methods with respect to changes of the hyperparameters.

The analysis of the mean areas under curves (Fig. 8) suggests that the hyperparameters almost do not influence on the results of the EM learning algorithm, while there is a significant dependence between hyperparameters' changes and results of the VB and GS learning algorithms. These conclusions are confirmed by examination of the individual runs of the algorithms. For example, Fig. 9 presents the precision-recall curves for all 20 runs with different initializations of four methods for the QMUL data: the VB learning algorithm using the plug-in approximation of the marginal likelihood with the prior types 1 and H of the hyperparameters' settings and the EM learning algorithm with the same prior groups of the hyperparameters' settings. One can notice that the variance of the curves for the VB learning algorithm with the prior type 1 is larger than the corresponding variance with the prior type H, while the similar variances for the EM learning algorithm are very close to each other.

Note that the results of the EM learning algorithm with the prior type 1 do not significantly differ from the results with the other priors, despite of the fact that the prior type 1 actually cancels out the prior influence on the parameters' estimates and equates the MAP and maximum likelihood estimates. We can conclude that the choice of the hyperparameters' settings is not a problem for the EM learning algorithm and we can even simplify the derivations considering only the maximum likelihood estimates without the prior influence.

The VB and GS learning algorithms require a proper choice of the hyperparameters' settings as they can significantly change the anomaly detection performance. This choice can be

[2]The computational time is provided for a laptop computer with i7-4702HQ CPU with 2.20GHz, 16 GB RAM using MATLAB R2015a implementation.

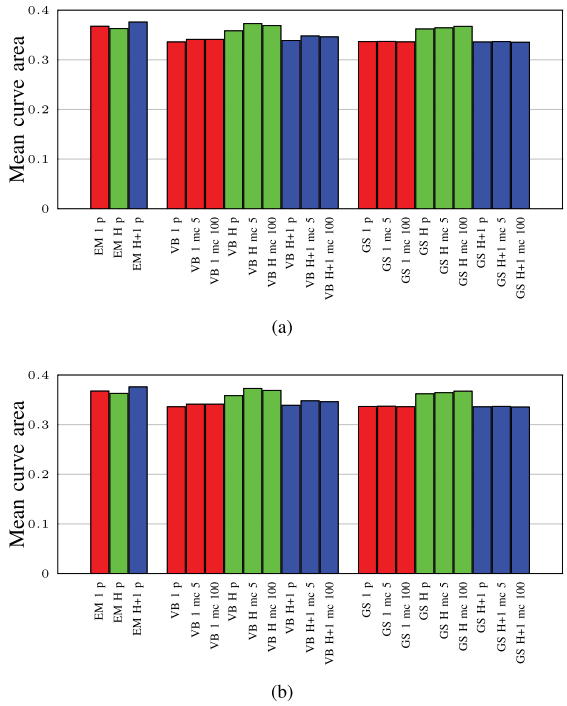


Fig. 8. Results of anomaly detection. (a) Mean areas under precision-recall curves for the QMUL data. (b) Mean areas under precision-recall curves for the Idiap data.

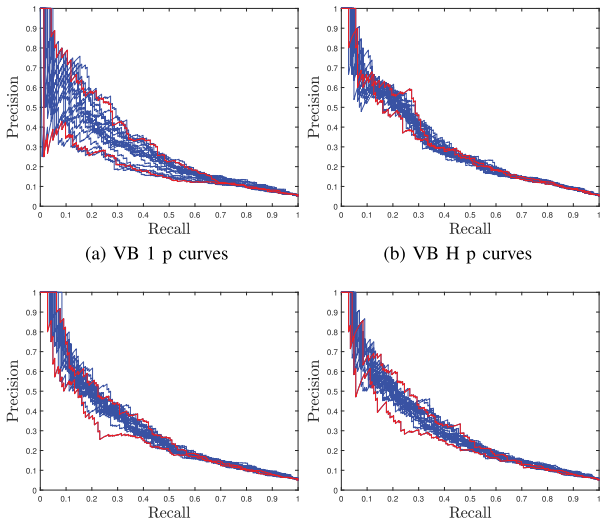


Fig. 9. Hyperparameters sensitivity of the precision-recall curves. Independent runs of the VB learning algorithm with (a) prior type 1 and (b) prior type H. Independent runs of the EM learning algorithm with (c) prior type 1 and (d) prior type H. The red color highlights the curves with the maximum and minimum areas under curves.

TABLE II
MEAN AREA UNDER PRECISION-RECALL CURVES

Dataset	EM	VB	GS
QMUL	0.3166	0.3155	0.2970
Idiap	0.3759	0.3729	0.3673

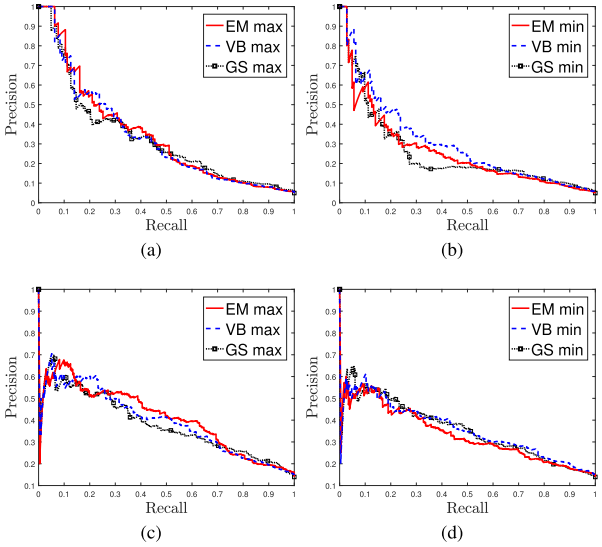


Fig. 10. Precision-recall curves with the maximum and minimum areas under curves for the three learning algorithms (maximum and minimum are among all the runs with different initializations for all groups of hyperparameters' settings and all types of marginal likelihood approximations). (a) "Best" curves for the QMUL data, i.e., the curves with the maximum area under a curve. (b) "Worst" curves for the QMUL data, i.e., the curves with the minimum area under a curve. (c) "Best" curves for the Idiap data. (d) "Worst" curves for the Idiap data.

performed empirically or with the type II maximum likelihood approach [28].

2) Marginal Likelihood Approximation Influence: In this section, the influence of the type of the marginal likelihood approximation on the anomaly detection results is studied.

The average results for both data sets (Fig. 8) demonstrate that the type of the marginal likelihood approximation does not influence remarkably on anomaly detection performance. As the plug-in approximation requires less computational resources both in terms of time and memory (as there is no need to sample and store posterior samples and average among them), this type of approximation is recommended to be used for anomaly detection in the proposed framework.

3) Learning Algorithms Comparison: This section compares the anomaly detection performance obtained by three learning algorithms.

The best results in terms of a mean area under a precision-recall curve are obtained by the EM learning algorithm, the worst results are obtained by the GS learning algorithm (Fig. 8 and Table II). In Table II, for each learning algorithm the group of hyperparameters' settings and the type of marginal likelihood approximation is chosen to have the maximum of the mean area under curves, where a mean is taken over independent runs of the same method and maximum is taken among different settings for the same learning algorithm.

Fig. 10 presents the best and the worst precision-recall curves (in terms of the area under them) for the individual runs of the learning algorithms. The figure shows that among the individual runs the EM learning algorithm also demonstrates the most accurate results. Although, the minimum area under the precision-recall curve for the EM learning algorithm is less than the area under the corresponding curve for the VB algorithm. It means that the variance among the individual curves for the EM learning algorithm is larger in comparison with the VB learning algorithm.

The variance of the precision-recall curves for both VB and GS learning algorithms is relatively small. However, the VB learning algorithm has the curves higher than the curves obtained by the GS learning algorithm. It can be confirmed by examination of the best and worst precision-recall curves (Fig. 10) and the mean values of the area under curves (Fig. 8 and Table II).

We also present the results of classification accuracy, i.e., the fraction of the correctly classified documents, for anomaly detection, which can be achieved with some fixed threshold. The best classification accuracy for the EM learning algorithm in both data sets can be found in Table III.

4) Anomaly Localization: We apply the proposed method for anomaly localization, presented in Section V-B, and get promising results. We demonstrate the localization results for

TABLE III
BEST CLASSIFICATION ACCURACY FOR THE EM LEARNING ALGORITHM

Dataset	Accuracy
QMUL	0.9544
Idiap	0.8891

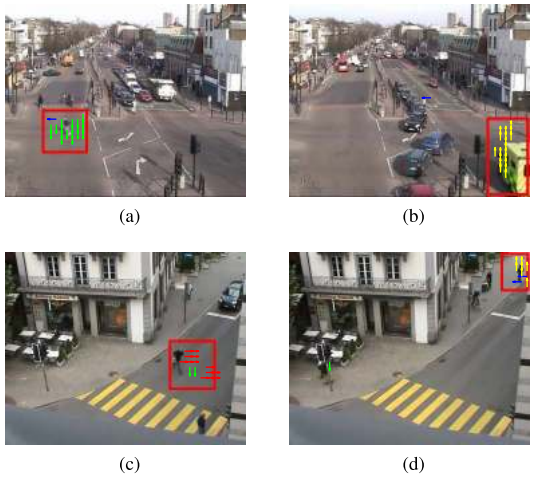


Fig. 11. Example of anomalies localization. The red rectangle is the manual localization. The arrows represent the visual words with the smallest marginal likelihood, the locations of the arrows are the results of the algorithmic anomaly localization. (a) and (b) Examples of anomaly localization for the QMUL data. (c) and (d) Samples of anomaly localization for the Idiap data.

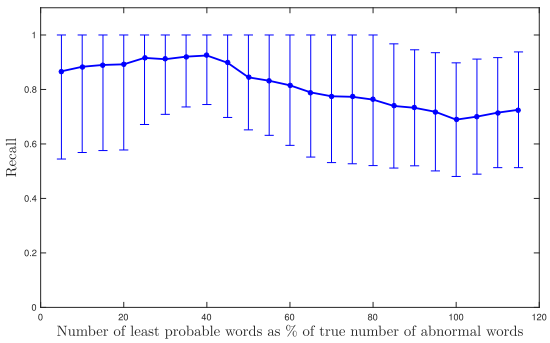


Fig. 12. Recall results of the proposed anomaly localization procedure.

the EM learning algorithm with the prior type H + 1 on both data sets in Fig. 11. The red rectangle is manually set to locate the abnormal events within the frame, the arrows correspond to the visual words with the smallest marginal likelihood computed by the algorithm. It can be seen that the abnormal events correctly localized by the proposed method.

For quantitative evaluation, we analyze 10 abnormal events (5 from each data set). For each clip, for a given number N_{top} of the least probable words, we measure the recall: recall = (TP/N_{an}), where N_{an} is the maximum possible number of abnormal words among N_{top}, i.e., $N_{\text{an}} = N_{\text{top}}$ if $N_{\text{top}} \leq N_{\text{total an}}$, where $N_{\text{total an}}$ is the total number of abnormal words, and $N_{\text{an}} = N_{\text{total an}}$ if $N_{\text{top}} > N_{\text{total an}}$. Fig. 12 presents the mean results for all events. One can notice, for example, that when the localization procedure can possibly detect 45% of the total number of abnormal words, it correctly finds $\approx 90\%$ of them.

VII. Conclusion

This paper presents two learning algorithms for the dynamic topic model for behavior analysis in video: the EM algorithm is developed for the MAP estimates of the model parameters and a VB inference algorithm is developed for calculating the posterior distributions of them. A detailed comparison of these proposed learning algorithms with the GS-based algorithm developed in [21] is presented. The differences and the similarities of the theoretical aspects for all three learning algorithms are well emphasized. The empirical comparison is performed for abnormal behavior detection using two unlabeled real video data sets. Both proposed learning algorithms demonstrate more accurate results than the algorithm proposed in [21] in terms of anomaly detection performance.

The EM learning algorithm demonstrates the best results in terms of the mean values of the performance measure, obtained by the independent runs of the algorithm with different random initializations. Although it is noticed that the variance among the precision-recall curves of the individual runs is relatively high, the VB learning algorithm shows the smaller variance among the precision-recall curves than the EM algorithm. The results show that the VB algorithm answers are more robust to different initialization values. However, it is shown that the results of the algorithm are significantly influenced by the choice of the hyperparameters. The hyperparameters require additional tuning before the algorithm can be applied to data. Note that the results of the EM learning algorithm only slightly depend on the choice of the hyperparameters' settings. Moreover, the hyperparameters can be even set in such a way as the EM algorithm is applied to obtain the maximum likelihood estimates instead of the MAP ones. Both the proposed learning algorithms—EM and VB—provide more accurate results in comparison to the GS-based algorithm.

We also demonstrate that consideration of marginal likelihoods of visual words rather than visual documents can provide satisfactory results about locations of anomalies within a frame. To our best knowledge, the proposed localization procedure is the first general approach in probabilistic topic modeling that requires only the presence of spatial information encoded in visual words.

Appendix A
EM Algorithm Derivations

This appendix presents the details of the proposed EM learning algorithm derivation. The objective function in the EM algorithm is

$$\begin{aligned}
&\mathcal{Q}(\boldsymbol{\Omega}, \boldsymbol{\Omega}^{\text{old}}) + \log p(\boldsymbol{\Omega}|\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\eta}, \boldsymbol{\gamma}) \\
&= \sum_{\mathbf{y}_{1:T_{tr}}} \sum_{\mathbf{z}_{1:T_{tr}}} (p(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{Old}}) \\
&\qquad \times \log p(\mathbf{x}_{1:T_{tr}}, \mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}|\boldsymbol{\Omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\eta})) \\
&\quad + \log p(\boldsymbol{\Omega}|\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\eta}, \boldsymbol{\gamma}) \\
&= \text{Const} + \sum_{z_1 \in \mathcal{Z}} (\log \pi_{z_1}\ p(z_1|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{Old}})) \\
&\quad + \sum_{t=2}^{T_{tr}} \sum_{z_t \in \mathcal{Z}} \sum_{z_{t-1} \in \mathcal{Z}} (\log \xi_{z_t, z_{t-1}}\ p(z_t, z_{t-1}|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{Old}})) \\
&\quad + \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} \sum_{y_{i,t} \in \mathcal{Y}} (\log \phi_{x_{i,t}, y_{i,t}}\ p(y_{i,t}|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{Old}})) \\
&\quad + \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} \sum_{z_t \in \mathcal{Z}} \sum_{y_{i,t} \in \mathcal{Y}} (\log \theta_{y_{i,t}, z_t}\ p(y_{i,t}, z_t|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{Old}})) \\
&\quad + \sum_{z \in \mathcal{Z}} (\eta_z - 1) \log \pi_z + \sum_{z \in \mathcal{Z}} \sum_{z' \in \mathcal{Z}} (\gamma_z - 1) \log \xi_{z,z'} \\
&\quad + \sum_{z \in \mathcal{Z}} \sum_{y \in \mathcal{Y}} (\alpha_y - 1) \log \theta_{y,z} + \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} (\beta_x - 1) \log \phi_{x,y}.
\end{aligned} \tag{44}$$

On the M-step, the function (44) is maximized with respect to the parameters $\boldsymbol{\Omega}$ with fixed values for $p(z_1|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{Old}})$, $p(z_t, z_{t-1}|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{Old}})$, $p(y_{i,t}|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{Old}})$, $p(y_{i,t}, z_t|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{Old}})$. The optimization problem can be solved separately for each parameter, which leads to (6)–(8).

On the E-step, for the efficient implementation, the forward–backward steps are developed for the auxiliary variables $\acute{\alpha}_z(t)$ and $\acute{\beta}_z(t)$

$$\begin{aligned}
\acute{\alpha}_z(t) &\stackrel{\text{def}}{=} p(\mathbf{x}_1, \ldots, \mathbf{x}_t, z_t = z|\boldsymbol{\Omega}^{\text{Old}}) \\
&= \sum_{\mathbf{z}_{1:t-1}} \pi_{z_1}^{\text{Old}} \left[\prod_{\acute{t}=2}^{t-1} \xi_{z_{\acute{t}}, z_{\acute{t}-1}}^{\text{Old}}\right] \left[\prod_{\acute{t}=1}^{t-1} \prod_{i=1}^{N_{\acute{t}}} \sum_{y \in \mathbf{Y}} \phi_{x_{i,\acute{t}}, y}^{\text{Old}} \theta_{y, z_{\acute{t}}}^{\text{Old}}\right] \\
&\quad \times \xi_{z_t = k, z_{t-1}}^{\text{Old}} \prod_{i=1}^{N_t} \sum_{y \in \mathcal{Y}} \phi_{x_{i,t}, y}^{\text{Old}} \theta_{y, z_t = z}^{\text{Old}}.
\end{aligned} \tag{45}$$

Reorganization of the terms in (45) leads to the recursive expressions (10).

Similarly for $\acute{\beta}_z(t)$

$$\begin{aligned}
\acute{\beta}_k(t) &\stackrel{\text{def}}{=} p(\mathbf{x}_{t+1}, \ldots, \mathbf{x}_{T_{tr}}|z_t = z, \boldsymbol{\Omega}^{\text{Old}}) \\
&= \sum_{\mathbf{z}_{t+1:T_{tr}}} \xi_{z_{t+1}, z_t = z}^{\text{Old}} \left[\prod_{\acute{t}=t+2}^{T_{tr}} \xi_{z_{\acute{t}}, z_{\acute{t}-1}}^{\text{Old}}\right] \\
&\quad \times \prod_{\acute{t}=t+1}^{T_{tr}} \prod_{i=1}^{N_{\acute{t}}} \sum_{y \in \mathcal{Y}} \phi_{x_{i,\acute{t}}, y}^{\text{Old}} \theta_{y, z_{\acute{t}}}^{\text{Old}}.
\end{aligned} \tag{46}$$

The recursive formula (11) is obtained by interchanging the terms in (46).

The required posterior of the hidden variables terms $p(z_1|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{Old}})$, $p(z_t, z_{t-1}|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{Old}})$,

$p(y_{i,t}|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{Old}})$, $p(y_{i,t}, z_t|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}^{\text{Old}})$ are then expressed via the axillary variables $\acute{\alpha}_z(t)$ and $\acute{\beta}_z(t)$, which leads to (13)–(16).

Appendix B
VB Algorithm Derivations

This appendix presents the details of the proposed VB inference derivation. We have separated the parameters and the hidden variables. Let us consider the update formula of the VB inference scheme [28] for the parameters

$$\begin{aligned}
&\log q(\boldsymbol{\Omega}) \\
&= \text{Const} + \mathbb{E}_{q(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}})} \log p(\mathbf{x}_{1:T_{tr}}, \mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}, \boldsymbol{\Omega}|\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
&= \text{Const} + \mathbb{E}_{q(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}})} \Bigg(\sum_{z\in\mathcal{Z}} (\eta_z - 1) \log \pi_z \\
&\quad + \sum_{z\in\mathcal{Z}} \sum_{\tilde{z}\in\mathcal{Z}} (\gamma_{\tilde{z}} - 1) \log \xi_{\tilde{z},z} + \sum_{z\in\mathcal{Z}} \sum_{y\in\mathcal{Y}} (\alpha_y - 1) \log \theta_{y,z} \\
&\quad + \sum_{y\in\mathcal{Y}} \sum_{x\in\mathcal{X}} (\beta_x - 1) \log \phi_{x,y} + \sum_{z\in\mathcal{Z}} \mathbb{I}(z_1 = z) \log \pi_z \\
&\quad + \sum_{t=2}^{T_{tr}} \sum_{z\in\mathcal{Z}} \sum_{\tilde{z}\in\mathcal{Z}} \mathbb{I}(z_t = \tilde{z}) \mathbb{I}(z_{t-1} = z) \log \xi_{\tilde{z},z} \\
&\quad + \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} \sum_{y\in\mathcal{Y}} \mathbb{I}(y_{i,t} = y) \log \phi_{x_{i,t},y} \\
&\quad + \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} \sum_{z\in\mathcal{Z}} \sum_{y\in\mathcal{Y}} \mathbb{I}(y_{i,t} = y) \mathbb{I}(z_t = z) \log \theta_{y,z} \Bigg). \quad (47)
\end{aligned}$$

One can notice that $\log q(\boldsymbol{\Omega})$ is further factorized as in (18). Now each factorization term can be considered independently. Derivations of (19)–(22) are very similar to each other. We provide the derivation only of the term $q(\boldsymbol{\Phi})$

$$\begin{aligned}
\log q(\boldsymbol{\Phi}) &= \text{Const} + \mathbb{E}_{q(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}})} \Bigg(\sum_{y\in\mathcal{Y}} \sum_{x\in\mathcal{X}} (\beta_x - 1) \log \phi_{x,y} \\
&\qquad + \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} \sum_{y\in\mathcal{Y}} \mathbb{I}(y_{i,t} = y) \log \phi_{x_{i,t},y} \Bigg) \\
&= \text{Const} + \sum_{y\in\mathcal{Y}} \sum_{x\in\mathcal{X}} (\beta_x - 1) \log \phi_{x,y} \\
&\quad + \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} \sum_{y\in\mathcal{Y}} \log \phi_{x_{i,t},y} \underbrace{\mathbb{E}_{q(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}})} (\mathbb{I}(y_{i,t} = y))}_{q(y_{i,t}=y)} \\
&= \text{Const} + \sum_{y\in\mathcal{Y}} \sum_{x\in\mathcal{X}} \log \phi_{x,y} \\
&\quad \times \Bigg(\beta_x - 1 + \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} \mathbb{I}(x_{i,t} = x) q(y_{i,t} = y) \Bigg). \quad (48)
\end{aligned}$$

It can be noticed from (48) that the distribution of $\boldsymbol{\Phi}$ is a product of the Dirichlet distributions (19).

The update formula in the VB inference scheme for the hidden variables is as follows:

$$\begin{aligned}
&\log q(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}) \\
&= \text{Const} + \mathbb{E}_{q(\boldsymbol{\pi}) q(\boldsymbol{\Xi}) q(\boldsymbol{\Theta}) q(\boldsymbol{\Phi})} \log p(\mathbf{x}_{1:T_{tr}}, \mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}, \\
&\qquad\qquad \boldsymbol{\Omega}|\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
&= \text{Const} + \sum_{z\in\mathcal{Z}} \mathbb{I}(z_1 = z) \mathbb{E}_{q(\boldsymbol{\pi})} \log \pi_z \\
&\quad + \sum_{t=2}^{T_{tr}} \sum_{z\in\mathcal{Z}} \sum_{\tilde{z}\in\mathcal{Z}} \mathbb{I}(z_t = \tilde{z}) \mathbb{I}(z_{t-1} = z) \mathbb{E}_{q(\boldsymbol{\Xi})} \log \xi_{\tilde{z},z} \\
&\quad + \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} \sum_{y\in\mathcal{Y}} \mathbb{I}(y_{i,t} = y) \mathbb{E}_{q(\boldsymbol{\Phi})} \log \phi_{x_{i,t},y} \\
&\quad + \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} \sum_{z\in\mathcal{Z}} \sum_{y\in\mathcal{Y}} \mathbb{I}(y_{i,t} = y) \mathbb{I}\ (z_t = z) \mathbb{E}_{q(\boldsymbol{\Theta})} \log \theta_{y,z}. \quad (49)
\end{aligned}$$

We know from the parameters update (19)–(22) that their distributions are Dirichlet. Therefore, $\mathbb{E}_{q(\boldsymbol{\pi})} \log \pi_z = \psi(\tilde{\eta}_z) - \psi(\sum_{z'\in\mathcal{Z}} \tilde{\eta}_{z'})$ and similarly for all the other expected value expressions.

Using the introduced notations (23)–(26), the update formula (49) for the hidden variables can be then expressed as

$$\begin{aligned}
&\log q(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}) \\
&\quad = \text{Const} + \sum_{z\in\mathcal{Z}} \mathbb{I}(z_1 = z) \log \tilde{\pi}_z \\
&\qquad + \sum_{t=2}^{T_{tr}} \sum_{z\in\mathcal{Z}} \sum_{\tilde{z}\in\mathcal{Z}} \mathbb{I}(z_t = \tilde{z}) \mathbb{I}(z_{t-1} = z) \log \tilde{\xi}_{\tilde{z},z} \\
&\qquad + \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} \sum_{y\in\mathcal{Y}} \mathbb{I}(y_{i,t} = y) \log \tilde{\phi}_{x_{i,t},y} \\
&\qquad + \sum_{t=1}^{T_{tr}} \sum_{i=1}^{N_t} \sum_{z\in\mathcal{Z}} \sum_{y\in\mathcal{Z}} \mathbb{I}(y_{i,t} = y) \mathbb{I}(z_t = z) \log \tilde{\theta}_{y,z}. \quad (50)
\end{aligned}$$

The approximated distribution of the hidden variables is then

$$\begin{aligned}
&q(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}) \\
&\qquad = \frac{1}{\tilde{K}} \tilde{\pi}_{z_1} \left[\prod_{t=2}^{T_{tr}} \tilde{\xi}_{z_t, z_{t-1}} \right] \prod_{t=1}^{T_{tr}} \prod_{i=1}^{N_t} \tilde{\phi}_{x_{i,t}, y_{i,t}} \tilde{\theta}_{y_{i,t}, z_t} \quad (51)
\end{aligned}$$

where $\tilde{K}$ is a normalization constant. Note that the expression of the true posterior distribution of the hidden variables is the same up to replacing the true parameters' variables with the corresponding tilde variables

$$\begin{aligned}
&p(\mathbf{y}_{1:T_{tr}}, \mathbf{z}_{1:T_{tr}}|\mathbf{x}_{1:T_{tr}}, \boldsymbol{\Omega}) \\
&\qquad = \frac{1}{K} \pi_{z_1} \left[\prod_{t=2}^{T_{tr}} \xi_{z_t, z_{t-1}} \right] \prod_{t=1}^{T_{tr}} \prod_{i=1}^{N_t} \phi_{x_{i,t}, y_{i,t}} \theta_{y_{i,t}, z}. \quad (52)
\end{aligned}$$

Therefore, to compute the required expressions of the hidden variables $q(z_1 = z)$, $q(z_{t-1} = z, z_t = z')$, $q(y_{i,t} = y, z_t = z)$, and $q(y_{i,t} = y)$ one can use the same forward–backward procedure and update formula as in the E-step of the EM algorithm replacing all the parameter variables with the corresponding introduced tilde variables.

REFERENCES

[1] R. Raghavendra, A. Del Bue, M. Cristani, and V. Murino, "Optimizing interaction force for global anomaly detection in crowded scenes," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 136–143.

[2] S.-H. Yen and C.-H. Wang, "Abnormal event detection using HOSF," in *Proc. Int. Conf. IT Converg. Secur. (ICITCS)*, Dec. 2013, pp. 1–4.

[3] K. Ouivirach, S. Gharti, and M. N. Dailey, "Incremental behavior modeling and suspicious activity detection," *Pattern Recognit.*, vol. 46, no. 3, pp. 671–680, 2013.

[4] G. Zhou, A. Cichocki, Y. Zhang, and D. P. Mandic, "Group component analysis for multiblock data: Common and individual feature extraction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2426–2439, Nov. 2016.

[5] Z. Su, H. Wei, and S. Wei, "Crowd event perception based on spatiotemporal Weber field," *J. Elect. Comput. Eng.*, vol. 2014, Jan. 2014, Art. no. 719810, doi: 10.1155/2014/719810.

[6] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, "Abnormal crowd behavior detection and localization using maximum sub-sequence search," in *Proc. 4th ACM/IEEE Int. Workshop Anal. Retr. Tracked Events Motion Imag. Stream (ARTEMIS)*, New York, NY, USA, 2013, pp. 49–58.

[7] Y.-H. Liu, Y.-C. Liu, and Y.-J. Chen, "Fast support vector data descriptions for novelty detection," *IEEE Trans. Neural Netw.*, vol. 21, no. 8, pp. 1296–1313, Aug. 2010.

[8] L. Maddalena and A. Petrosino, "Stopped object detection by learning foreground model in videos," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 723–735, May 2013.

[9] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations," *J. Comput. Phys.*, vol. 79, no. 1, pp. 12–49, 1988.

[10] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "Novelty detection using level set methods," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 576–588, Mar. 2015.

[11] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 555–560, Mar. 2008.

[12] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[13] L. Kratz and K. Nishino, "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1446–1453.

[14] C. Brighenti and M. Á. Sanz-Bobi, "Auto-regressive processes explained by self-organized maps. Application to the detection of abnormal behavior in industrial processes," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2078–2090, Dec. 2011.

[15] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, New York, NY, USA, 1999, pp. 50–57.

[16] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[17] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 935–942.

[18] J. Li, S. Gong, and T. Xiang, "Global behaviour inference using probabilistic latent semantic analysis," in *Proc. Brit. Mach. Vis. Conf.*, 2008, pp. 193–202.

[19] J. Varadarajan and J. M. Odobez, "Topic models for scene analysis and abnormality detection," in *Proc. 12th IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Sep. 2009, pp. 1338–1345.

[20] X. Wang, X. Ma, and W. E. L. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 539–555, Mar. 2009.

[21] T. Hospedales, S. Gong, and T. Xiang, "Video behaviour mining using a dynamic topic model," *Int. J. Comput. Vis.*, vol. 98, no. 3, pp. 303–323, 2012.

[22] H. Jeong, Y. Yoo, K. M. Yi, and J. Y. Choi, "Two-stage online inference model for traffic pattern analysis and anomaly detection," *Mach. Vis. Appl.*, vol. 25, no. 6, pp. 1501–1517, 2014.

[23] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 1, pp. 5228–5235, 2004.

[24] J.-T. Chien and M.-S. Wu, "Adaptive Bayesian latent semantic analysis," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 1, pp. 198–207, Jan. 2008.

[25] O. Isupova, L. Mihaylova, D. Kuzin, G. Markarian, and F. Septier, "An expectation maximisation algorithm for behaviour analysis in video," in *Proc. 18th Int. Conf. Inf. Fusion (Fusion)*, Jul. 2015, pp. 126–133.

[26] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. B (Methodol.)*, vol. 39, no. 1, pp. 1–38, 1977.

[27] K. Vorontsov and A. Potapenko, "Additive regularization of topic models," *Mach. Learn.*, vol. 101, pp. 303–323, Oct. 2015.

[28] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[29] I. M. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, no. 2, pp. 183–233, 1999.

[30] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh, "On smoothing and inference for topic models," in *Proc. 25th Conf. Uncertainty Artif. Intell. (UAI)*, 2009, pp. 27–34.

[31] T. S. F. Haines and T. Xiang, "Video topic modelling with behavioural segmentation," in *Proc. 1st ACM Int. Workshop Multimodal Pervasive Video Anal. (MPVA)*, New York, NY, USA, 2010, pp. 53–58.

[32] D. Pathak, A. Sharang, and A. Mukerjee, "Anomaly localization in topic-based analysis of surveillance videos," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 389–395.

Olga Isupova received the Specialist (equal to M.Sc.) degree in applied mathematics and computer science from Lomonosov Moscow State University, Moscow, Russia, in 2012. She is currently pursuing the Ph.D. degree with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U.K.

She is currently an Early Stage Researcher of the FP7 Program TRAX with the University of Sheffield. Her current research interests include machine learning, Bayesian nonparametrics, and anomaly detection.

Danil Kuzin received the Specialist degree in applied mathematics and computer science from Lomonosov Moscow State University, Moscow, Russia, in 2012. He is currently pursuing the Ph.D. degree with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U.K.

He is currently an Engineer with Rinicom, Ltd., Lancaster, U.K. His current research interests include sparse modeling for video, nonparametric Bayes, and deep reinforcement learning.

Lyudmila Mihaylova (M'98–SM'08) is currently a Professor of signal processing and control with the Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U.K. Her current research interests include machine learning and autonomous systems with various applications such as navigation, surveillance, and sensor network systems.

Prof. Mihaylova is an Associate Editor of the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS and the *Elsevier Signal Processing Journal*. She was elected as a President of the International Society of Information Fusion (ISIF) in 2016. She is on the Board of Directors of ISIF. She was the General Co-Chair of the IET Data Fusion & Target Tracking 2014 and 2012 Conferences, a Program Co-Chair of the 19th International Conference on Information Fusion, 2016, and the Academic Chair of the Fusion 2010 Conference. She has given a number of talks and tutorials, including the plenary talk for the IEEE Sensor Data Fusion 2015 (Germany), the invited talks at the University of California, Los Angeles, CA, USA, the IPAMI Traffic Workshop 2016 (USA), and the IET ICWMMN 2013 in Beijing, China.