Series title:
Master graduation paper, Electrical Engineering

Commisioned by professor	Group & chair	Date of final presentation	Report number
prof. dr. ir. A. (Bert) de Vries	Signal Processing Systems prof. dr. ir. J.W.M. (Jan) Bergmans BIASlab prof. dr. ir. A. (Bert) de Vries	30/04/2021	

Department of Electrical Engineering

Den Dolech 2, 5612 AZ Eindhoven
P.O. Box 513, 5600 MB Eindhoven
The Netherlands

https://www.tue.nl/en/

A probabilistic approach to situated acoustic road event detection

by

Mark Johan Willem Legters

Internal supervisors: Bart van Erp, MSc
Albert Podusenko, MSc

External supervisors: Rik Sweep, MSc

A Probabilistic Approach to Situated Acoustic Road Event Detection

M.J.W. Legters
m.j.w.legters@student.tue.nl, ID 0816149

***Abstract*—We address the problem of situated acoustic road event detection. We present a general probabilistic framework to detect and annotate acoustic events. The framework relies on two generative probabilistic models – a "good" model that will learn the characteristic on-the-spot ("situated") acoustic dynamics, and a baseline model – on which we run signal processing tasks. We define the involved signal processing stages in the approach (parameter estimation, state estimation, and model comparison) as probabilistic inference tasks. We introduce generative models focusing on the characteristic acoustics on the road, allowing us to validate the framework for acoustic road event detection specifically. Our experimental results show that we are able to detect acoustic road events such as a crash and tire slipping using our presented probabilistic framework.**

***Index Terms*—Bayesian Machine Learning, Inference, Variational Free Energy, Factor Graphs, Sum-product Message Passing, Variational Message Passing, Bayesian Model Comparison, Acoustic Event Detection**

I. Introduction

ROAD traffic monitoring involves, amongst others, the detection of accidents or other hazardous events to quickly intervene with emergency teams and to guarantee the safety of people [1]. A reduction of time between an actual accident and emergency team dispatch decreases the mortality rate by approximately 6% [2], [3]. Many existing approaches to road traffic monitoring are based on visual information by means of surveillance cameras [4], [5], [6]. However, the use of cameras has its limitations, for example due to occlusions, the limited field of view of the camera, and varying illumination due to weather conditions and changes between day and night. Looking at those limitations, analysis of audio signals is a complementary tool. For example, the use of microphones and audio analysis is not affected by occlusions of varying illumination. Combining the two may improve the detection abilities of security systems [7].

Audio event detection can also be useful on its own. In very large areas or along big roads, using microphones instead of surveillance cameras can have positive impact on deployment costs of a security system [8], as microphones can cover a larger area while being cheaper than cameras [9]. In this work, we design a framework for the detection of audio events, in the context of monitoring road safety. Besides road accidents, which are registered and taken into account for safety figures, we also aim to detect abrupt maneuvers like tire slipping. Despite not ending up in safety figures, they are also an indicator of road safety.

Significant research effort has been put into audio analysis. Survey papers [10] and [11] show a comprehensive analysis of existing methods. Many methods are data-driven, which recently tend to focus on the use of neural networks. This paper focuses on a probabilistic, model-based approach. We present a general automated framework for on-the-spot acoustic event detection. The main idea of this framework is as follows. First, we propose a probabilistic generative signal model. Next, we fit this model to a situated acoustic environment. Once the model is fit to the acoustic environment, we use it to predict future observations based on past observations, and track its performance in doing so. We also track this acoustic signal against a baseline model, also resulting in a performance measure. Finally, we compare the models' performances to detect events.

We use a probabilistic approach because of its conceptual simplicity [12]. A model is supplied up front and consists of simple sparse building blocks. This both favors the interpretability of the model and results. Furthermore, probabilistic modeling incorporates uncertainty in a natural way. Also, there is no need for a large annotated data set with sufficient well-defined events upfront, as opposed to data-driven neural networks.

The main contributions of this paper are as follows. First, we present a general solution framework for the detection of acoustic events. We describe the stages in this framework, i.e. parameter estimation, online state estimation, and model comparison, as probabilistic inference tasks that can be executed automatically by means of message passing. Secondly, we validate this framework on a road application by defining required models and applying it to data relevant to this application.

The paper is organized as follows. In Section II we specify our problem and introduce the proposed solution framework. We specify the probabilistic models for our road event detection application in Section III. In these models, we use factor graphs and message passing to execute inference tasks, on which we elaborate in Section IV. Section V presents our experimental results of the proposed approach with the specified models. We postponed a discussion on related work to Section VI, as we deem this better fit in the context of our proposed solution. Next, Section VII reviews the results, discusses limitations of the proposed methodology, and gives directions for future work. Finally, Section VIII concludes the paper.

II. Problem statement and solution framework

The goal of this paper is to achieve automated on-the-spot acoustic road event detection using probabilistic generative

signal models. For this detection, we have a microphone at a certain fixed location along or above the road that delivers our acoustic signal. To achieve the goal, we propose a fully probabilistic solution framework consisting of several stages. These stages are (1) parameter estimation, (2) state estimation, and (3) model comparison.

In this research, we consider an event to be an occasion of unexpected deviation from characteristic behavior. In the scope of our road application, we consider characteristic behavior to be the sound of passing cars, which is typical to our application environment and does not need to be considered as an event. We need to take some expected deviation into account in the characteristic behavior, as not all cars produce the same sound. However, in the occasion of a crash or a slipping car, we no longer speak of characteristic behavior or an expected deviation from it; this behavior was not anticipated on, and therefore should be classified as an event.

During the first stage, we aim to capture the characteristic behavior mentioned in the previous paragraph. Using a generative probabilistic model and the recording of the characteristic behavior, in this case passing cars, we estimate the parameters θ that capture this characteristic behavior. This stage is visualized in Figure 1a. In the second stage, we fix the parameters that we estimated in the first stage in the model. Next, we run the model on microphone recordings that now not necessarily only contain characteristic behavior, but potentially also events we would like to detect. At the same time, an uninformed baseline model that is not tailored to the characteristic behavior is also run on the recording. This results in a performance score for both of the models. This stage is visualized in Figure 1b. In the final stage, model comparison (MC), the metrics of both models are compared, resulting in an output indicating whether an event was detected or not. This stage is visualized in Figure 1c. Each stage can be expressed as an inference task on a generative probabilistic model. We discussed each stage and its corresponding inference procedure in more detail below.

To define the aforementioned inference tasks representing the stages, we base ourselves on a general generative model, consisting of general random variables. Consider an observed signal x_n with time index $n = 1 \dots N$, modeled by dynamic latent state $\boldsymbol{s}_n$ and model parameters θ. Together, they form the general probabilistic generative model m_1, given as

$$p\left(x_{1:N},\ \boldsymbol{s}_{0:N},\ \theta\right) = \underbrace{p\left(\boldsymbol{s}_0\right) p\left(\theta\right)}_{\text{priors}} \prod_{n=1}^{N} \underbrace{p\left(\boldsymbol{s}_n \mid \boldsymbol{s}_{n-1}\right)}_{\text{state transition}} \underbrace{p\left(x_n \mid \boldsymbol{s}_n,\ \theta\right)}_{\text{likelihood}}. \quad (1)$$

Now that we defined a generative model, we can formulate all signal processing tasks as inference tasks on this generative model.

A. Stage 1: parameters estimation

In the first stage of our framework, the parameter estimation stage, we aim to capture the expected behavior of our environment: the passing cars on the road. We have made an hypothesis on this expected behavior in the form of our model. We perform parameter estimation on the actual location where the event detection should take place so that we tailor the parameters to the specific location. Based on a recording, the model parameters θ that capture this on-the-spot behavior are estimated through probabilistic inference on the model. This inference task calculates the posterior probability of the parameters θ, given the recording $x_{1:N}$. The inference task boils down to calculating

$$p\left(\theta \mid x_{1:N}\right) \propto p\left(\theta\right) \int p\left(\boldsymbol{s}_0\right) \prod_{n=1}^{N} p\left(\boldsymbol{s}_n \mid \boldsymbol{s}_{n-1}\right) p\left(x_n \mid \boldsymbol{s}_n, \theta\right) \mathrm{d}\boldsymbol{s}_{0:N}. \quad (2)$$

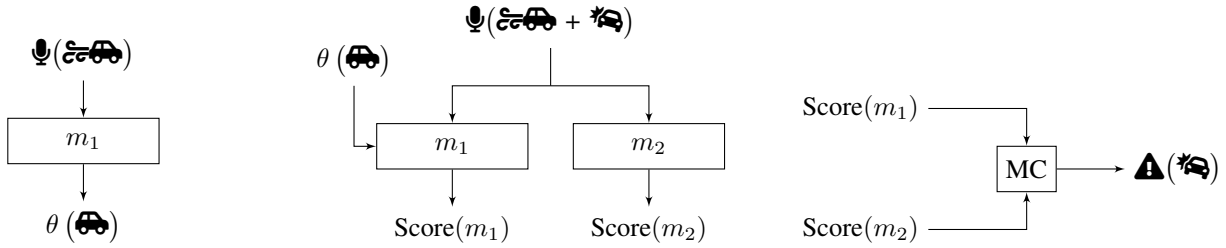
B. Stage 2: state estimation

During state estimation, which is the second stage, we use our model from (1) and update our model prior using the learned parameters from (2). During state estimation, we are computing estimates for the next state in the model in an online, continuous fashion. At the same time, we compute an evaluation of the current model performance. In order to achieve the state estimates, i.e. computing the posterior $p\left(\boldsymbol{s}_n \mid x_{1:n}\right)$, we apply Bayes rule [13]. Also, we take the model definition (1) into account, along with its recursive nature. This sequential Bayesian updating leads to the posterior state update equation, which makes use of the Chapman-Kolmogorov integral [14]. This inference task is given by

$$\underbrace{p\left(\boldsymbol{s}_n \mid x_{1:n}\right)}_{\text{posterior}} \underbrace{p\left(x_n \mid x_{1:n-1}\right)}_{\text{evidence}} = \underbrace{p\left(x_n \mid \boldsymbol{s}_n\right)}_{\text{likelihood}} \int \underbrace{p\left(\boldsymbol{s}_n \mid \boldsymbol{s}_{n-1}\right)}_{\text{state transition}} \underbrace{p\left(\boldsymbol{s}_{n-1} \mid x_{1:n-1}\right)}_{\text{prior}} \mathrm{d}\boldsymbol{s}_{n-1}. \quad (3)$$

In (3), the terms on the right-hand side are given after x_n has been observed; the state transition and the likelihood are a result of the defined model. The prior is inferred recursively based on initial prior $\boldsymbol{s}_0$, as the posterior of the previous time step serves as prior for the current time step. The task here is to compute the terms on the left-hand side. The posterior includes the estimate for the next state in the form of a probability distribution, and therefore fulfills the task of the second stage. The other left-hand side term is the model evidence, a scalar value that serves as a measure for how well the model predicts the current observation, based on past observations. While estimating the states, we aim to keep track of the evidence as a metric of how well the model performs on the data, as we need this for the model comparison stage.

While the evidence can be recursively updated in the case of simple models, like linear Gaussian models [14, p. 57], in more complex models, the involved integrals in the inference task can become intractable due to non-conjugate prior-posterior pairing, for example. To circumvent this intractability, we use



(a) Stage 1: parameters estimation (b) Stage 2: state estimation (c) Stage 3: model comparison

Figure 1. A schematic overview of the proposed solution framework, showing the three stages, being parameter estimation (a), state estimation (b), and model comparison (c). During parameter estimation, the parameters θ of the main model m_1 that describe the characteristic behavior of the environment are estimated. During state estimation, the main model m_1 (with the estimated parameters θ) and a baseline model m_2 are run on the input data, resulting in a score of the performance of both models. During the model comparison (MC) stage, the performance of both models is compared, outputting whether an event is detected or not.

an approximate distribution for the latent states $\boldsymbol{s}_n$ instead. Consider the variational free energy functional, given by

$$F_n[q] \triangleq \int q(\boldsymbol{s}_n) \log \frac{q(\boldsymbol{s}_n)}{p(\boldsymbol{s}_n, x_n \mid x_{1:n-1})} \mathrm{d}\boldsymbol{s}_n \tag{4a}$$

$$= \underbrace{-\log p(x_n \mid x_{1:n-1})}_{\text{log-evidence}} + \underbrace{\int q(\boldsymbol{s}_n) \log \frac{q(\boldsymbol{s}_n)}{p(\boldsymbol{s}_n \mid x_{1:n})} \mathrm{d}\boldsymbol{s}_n}_{\text{KL divergence} \geq 0}, \tag{4b}$$

where $q(\boldsymbol{s}_n)$ is this approximation of the posterior distribution of the states. The variational free energy is an approximation, more specifically an upper bound, to the negative log-evidence. As the KL-divergence term in (4) is non-negative by definition and only equals zero if $q(\boldsymbol{s}_n) = p(\boldsymbol{s}_n \mid x_{1:n})$, and the log-evidence term in (4) is independent of $q(\boldsymbol{s})$, minimization of $F_n[q]$ with respect to q leads to approximations for both the evidence and the posterior state distribution, which are the unknowns in (3).

C. Stage 3: model comparison

We have now a learned model for passing cars and apply it to the data via state estimation, while also keeping track of (the approximation of) the evidence as a performance measure. It is now time to perform the event detection.

In a fully Bayesian approach, model performance is evaluated through Bayesian model comparison, the Bayesian version of hypothesis testing. In stage 3, the model comparison involves computing the ratio between the posterior model probability for a certain model and an alternative one. This gives us an indication of how well one model matches the data, relative to an alternative model.

In the scope of our application, ideally, we use the following two models. First, we have m_1 from (1) with the inferred parameters θ from the parameter estimation stage (2) as fixed parameters. This generative probabilistic model incorporates the characteristic acoustic behavior. Secondly, as an alternative model, we ideally have a signal model for events specifically. However, we do not have any specific information about the events that we want to detect. Any signal not matching the acoustic environment could potentially be an event. We are detecting anomalies in the environmental acoustic dynamics. We cannot model the potentially broad spectrum of anomalies.

To still benefit from the closed nature of the Bayesian model comparison approach, we do define a second model. This model specifies the same data $x_{1:N}$, but with different states $z_{1:N}$, and has a simple, mediocre nature. It is acting as a baseline model. This probabilistic generative baseline model m_2 is defined as

$$p(x_{1:N},\ z_{1:N},\ \theta). \tag{5}$$

On this model, we also apply the inference task (3), which besides the state updates again supplies us with the evidence. This allows us to make the model comparison and consequently draw conclusions, despite not having a specific event model. This comes down to computing

$$\underbrace{\frac{p(m_1 \mid x_{1:N})}{p(m_2 \mid x_{1:N})}}_{\text{posterior ratio}} = \underbrace{\frac{p(x_{1:N} \mid m_1)}{p(x_{1:N} \mid m_2)}}_{\text{evidence ratio}} \cdot \underbrace{\frac{p(m_1)}{p(m_2)}}_{\text{prior ratio}}, \tag{6}$$

where $p(m_u)$ is the prior model probability for models $u \in \{1, 2\}$, subject to $p(m_1) + p(m_2) = 1$. If the posterior probability ratio is larger than 1, i.e., m_1 outperforms m_2, we are dealing with characteristic behavior, or expected deviations thereof. However, in the case of a ratio smaller than 1, m_2 outperforms m_1. The baseline model then matches the data better than the complex model that incorporates the characteristic behavior. This means that the data unexpectedly deviates from characteristic behavior, and therefore we are dealing with an event at that point, according to our definition.

As we have mentioned, the involved integrals in the calculation of the evidence, which we need for inference task (6), can become intractable due to non-conjugate prior-posterior pairing, for example. To circumvent this intractability, we use the variational free energy introduced in (4) to approximate the evidence terms in (6). Using this approximation in combination with the fact that posterior model probabilities of both our models add up to one, we can rewrite (6) to

$$p(m_2 \mid x_{1:N}) \approx \left(1 + e^{F_1 - F_2 - \log \frac{p(m_1)}{p(m_2)}}\right)^{-1}, \tag{7}$$

where F_1 and F_2 represent the variational free energy in m_1 and m_2, respectively. $p(m_2 \mid x_{1:N})$ now represents the posterior model probability of m_2, hence the probability of an event, and is the embodiment of our event detection.

III. Model specification

The proposed methodology in Section II depends on two probabilistic generative signal models. In this section, we further specify both of these generative models. For these models, we consider the signal $\boldsymbol{x}_n = \left[x_n^1, x_n^2, \ldots, x_n^M\right]^\top \in \mathbb{R}^M$, where every $x \in \mathbb{R}$ represents a log-power spectrum coefficient, $n = 1, \ldots, N$ specifies the discrete time indices, and M is the amount of log-power spectrum coefficients per time step. First, we define the main model m_1, which captures the characteristic behavior of our application environment. Secondly, we give a specification of our baseline model m_2, which serves as the mediocre counterpart in the model comparison.

A. Main model

The main model captures the characteristic behavior of the environment. In our case, we choose to base our model on the physics of cars passing by. The model decomposes our observations into a characteristic sound profile on the one hand, and a dynamic gain on the other hand. We first describe the observation model as a function of this sound profile and gain, followed by specifying the modeling of this sound profile as well as the gain.

1) Observation model: In our observation model, we aim to model the physical situation at a fixed signal sensor when a car is passing by. We take into account two aspects here: the sound of a car, and the passing motion. We capture the characteristic sound of a car by means of a sound profile. A sound profile reflects the relative presence of frequencies in the sound over the spectrum, forming a characteristic footprint of this sound. As a car approaches, the sound will get louder, and it will dampen again once the car has passed. Louder sound means that signal amplitude has increased, while keeping the same frequency content. Therefore, we model the presence of motion relative to the sensor as a time-varying gain that multiplies the amplitude of the whole spectrum, i.e. the sound profile. Because we are working in the log-power domain, the multiplicative dependencies become additive. This leads to the likelihood function

$$p\left(\boldsymbol{x}_n \mid \boldsymbol{h}, g_n\right) = \mathcal{N}\left(\boldsymbol{x}_n \mid \boldsymbol{h} + g_n \mathbf{1}_M, \Lambda_x\right), \tag{8}$$

where $\boldsymbol{h} \in \mathbb{R}^M$ is the time-invariant sound profile, $g_n \in \mathbb{R}$ is the gain at time index n, $\mathbf{1}_M$ is a column vector of M ones, Λ_h is a diagonal precision matrix of the observation noise, and $\mathcal{N}\left(\cdot \mid \boldsymbol{\mu}, \Lambda\right)$ is the Gaussian distribution with mean vector $\boldsymbol{\mu}$ and precision matrix Λ. This observation model extends to an arbitrary number of coefficients M.

2) Sound profile: The goal of the sound profile is to model the characteristic sound pattern in terms of log-power spectrum coefficients. As the pattern will not exactly be the same for every car, we expect deviations, and therefore we want to retain uncertainty in this model. We model the sound profile $\boldsymbol{h}$ as a multivariate Gaussian using

$$p\left(\boldsymbol{h} \mid \boldsymbol{m}_{\boldsymbol{h}}, \boldsymbol{W}_{\boldsymbol{h}}\right) = \mathcal{N}\left(\boldsymbol{h} \mid \boldsymbol{m}_{\boldsymbol{h}}, \boldsymbol{W}_{\boldsymbol{h}}\right), \tag{9}$$

where $\boldsymbol{m} \in \mathbb{R}^M$ is a vector of means and $\boldsymbol{W} \in \mathbb{R}_+^{M \times M}$ is the precision matrix. We have chosen to denote a matrix with a bold capital letter, as we use regular capital letters for upper bounds on variables. If we would infer the parameters $\boldsymbol{m}_{\boldsymbol{h}}$ and $\boldsymbol{W}_{\boldsymbol{h}}$, this would lead to incorrect estimates of the precision that do not reflect the uncertainty of sound profile parameters we require. In order to retain the required uncertainty, we also treat the sound profile parameters $\boldsymbol{m}_{\boldsymbol{h}}$ and $\boldsymbol{W}_{\boldsymbol{h}}$ in (9) as random variables, using

$$p\left(\boldsymbol{m}_{\boldsymbol{h}}\right) = \mathcal{N}\left(\boldsymbol{m}_{\boldsymbol{h}} \mid \boldsymbol{\mu}, \Lambda\right) \tag{10a}$$

$$p\left(\boldsymbol{W}_{\boldsymbol{h}}\right) = \mathcal{W}\left(\boldsymbol{W}_{\boldsymbol{h}} \mid \boldsymbol{V}, \nu\right). \tag{10b}$$

Here, $\mathcal{W}\left(\cdot \mid \boldsymbol{V}, \nu\right)$ is the Wishart distribution with scale matrix $\boldsymbol{V}$ and ν degrees of freedom. This distribution forms a conjugate prior with the precision matrix of a Gaussian distribution in case of a fixed mean, allowing closed-form relations between prior and posterior distributions.

3) State transition: We use the gain g_n to represent the motion relative to the signal sensor. As a car approaches the sensor, the amplitude of the signal – and therefore the signal energy – will increase. Likewise, when a car has passed, the amplitude will decrease again. As this motion happens in a gradual manner, we assume this behavior to be a Markov chain. More specifically, in our model we assume the future states to follow a one-dimensional Gaussian random walk, limiting the amount of change in gain per time frame. It is therefore modeled as

$$p\left(g_n \mid g_{n-1}\right) = \mathcal{N}\left(g_n \mid g_{n-1}, \gamma_g\right), \tag{11}$$

where $\gamma_g \in \mathbb{R}_+$ is the precision of the process noise.

4) Model summary: In summary, the generative probabilistic model for the signal of log-power spectrum coefficients is given by the set of equations (8), (9), (10a), (10b), and (11), and explicitly by the joint probability distribution

$$p\left(\boldsymbol{x}_{1:N}, g_{0:N}, \boldsymbol{h}, \boldsymbol{\theta}_{\boldsymbol{h}}\right) = p\left(g_0\right) p\left(\boldsymbol{h} \mid \boldsymbol{\theta}_{\boldsymbol{h}}\right) p\left(\boldsymbol{\theta}_{\boldsymbol{h}}\right) \prod_{n=1}^{N} p\left(\boldsymbol{x}_n \mid \boldsymbol{h}, g_n\right) p\left(g_n \mid g_{n-1}\right), \tag{12}$$

where $\boldsymbol{\theta}_{\boldsymbol{h}} = \{\boldsymbol{m}_{\boldsymbol{h}}, \boldsymbol{W}_{\boldsymbol{h}}\}$ are the parameters of the sound profile model and $p\left(\boldsymbol{\theta}_{\boldsymbol{h}}\right) = p\left(\boldsymbol{m}_{\boldsymbol{h}}\right) p\left(\boldsymbol{W}_{\boldsymbol{h}}\right)$. This distribution matches the probabilistic generative model definition in (1).

B. Baseline model

For our baseline model, we use (1) as a starting point, but apply some simplifications. We omit the parameters, as well as the time dependence of the state, which leaves us with an observation model and a state. We will now shortly detail both.

The baseline model works with the same observations $x_{1:N}$ as the main model. We assume the observation to be a noisy version of the state. This leads to a likelihood function given by

$$p\left(\boldsymbol{x}_n \mid \boldsymbol{z}\right) = \mathcal{N}\left(\boldsymbol{x}_n \mid \boldsymbol{z}, \Lambda_x\right), \tag{13}$$

where $\boldsymbol{z} \in \mathbb{R}^M$ is the latent state, and $\Lambda_x \in \mathbb{R}_+^{M \times M}$ is a diagonal precision matrix for the observation noise.

The other simplification has been made regarding state, where we omit the time dependence. The model is fed with a

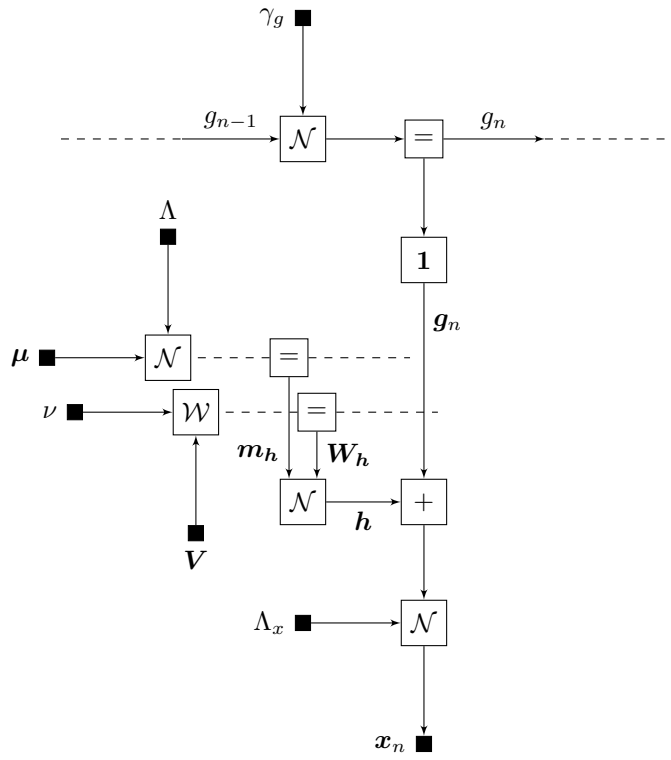


Figure 2. One time segment of the Forney-style factor graph representation of the model introduced in section III. The dashed edges indicate the temporal connections for extensibility of the graph in a similar manner for other time steps. The arrowheads indicate the generative direction of the model.

prior for the state on initialization, and this state is not updated after that. This results in a state model

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu_z}, \Lambda_{\boldsymbol{z}}), \tag{14}$$

where $\boldsymbol{\mu}_z \in \mathbb{R}^M$ is the mean vector, and $\Lambda_z \in \mathbb{R}_+^{M\times M}$ is the diagonal precision matrix. The idea behind this model is to create Gaussian distribution for each of the M log-power spectrum coefficients. Introducing small values in the diagonal precision matrix will lead to broad Gaussian distributions, resulting in a comparable mediocre behavior for a large range of observations. This is the requirement for the baseline model.

In summary, the probabilistic baseline model is given by equations (13) and (14), and explicitly defined by the joint probability distribution

$$p(\boldsymbol{x}_{1:N}, \boldsymbol{z}_{1:N}) = \prod_{n=1}^{N} p(\boldsymbol{z_n})\, p(\boldsymbol{x}_n \mid \boldsymbol{z_n}). \tag{15}$$

This concludes the specifications of general models m_1 and m_2 as defined in Section II. We use factor graphs to illustrate the sparse conditional dependencies between the random variables in our models. Figures 2 and 3 show the factor graph representation of our main model m_1 and baseline model m_2, respectively. In the next section, we will give a short overview of this type of probabilistic graphical model, and present an efficient inference methodology based on this representation.

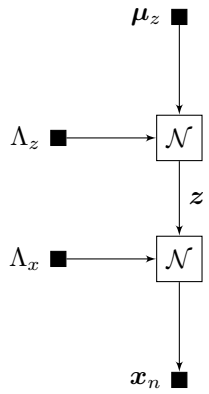


Figure 3. Forney-style factor graph representation of the model introduced in (15). The arrowheads indicate the generative direction of the model.

IV. Probabilistic inference

Now that we have defined the two models, we can use them in the proposed solution framework. In order to execute the parameter estimation and state estimation stages, we use the message passing paradigm on factor graphs as our probabilistic inference approach. Factor graphs serve as a graphical tool to visualize the probabilistic model, the inference tasks, and the message passing algorithm. We use a combination of message passing and factor graphs, as this allows an efficient, scalable, automatable, and modular probabilistic inference approach [15]. In this section, we present a short overview of factor graphs and message passing algorithms.

A. Forney-style Factor Graphs

Factor graphs are graphical models that represent the factorization of a global function. Within the context of this paper, the joint probability distributions specifying our generative models are these global functions. Specifically, in this paper, we focus on Forney-style factor graphs (FFGs). In these FFGs, every local function, or factor, is represented by a node. These nodes are interconnected by edges, representing random variables. An edge is connected to a node if and only if the random variable is an argument of the factor of this node [15], [16]. Due to limited conditional dependencies in the global function, the resulting factor graph has sparse connectivity.

Consider an example model $p(x_1, x_2, x_3, x_4, x_5)$ that we would like to represent using an FFG. This example model factorizes as

$$\begin{aligned} &p(x_1, x_2, x_3, x_4, x_5) = \\ &\qquad f_a(x_1, x_2)\, f_b(x_2, x_3, x_4)\, f_c(x_3)\, f_d(x_4, x_5), \end{aligned} \tag{16}$$

where the functions with alphabetical subscript indicate the factors within the model. From (16), the FFG can be constructed based on the three visualization rules of [15], as shown in Figure 4.

B. Sum-product message passing

Now suppose we want to calculate the marginal distribution $p(x_2)$ of (16). If the model could not be factorized, the

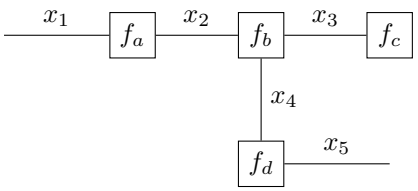


Figure 4. A Forney-style factor graph representation of (16).

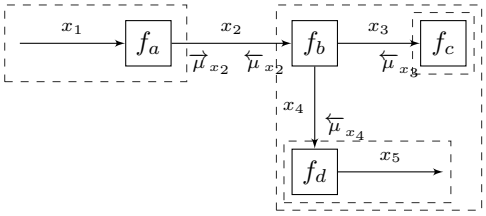


Figure 5. A Forney-style factor graph of (16), including the sum-product messages for calculation of the marginal distribution of x_2, as indicated in (18). In theory, an FFG is undirected. In this case, the edges have been directed to distinguish between forward and backward messages.

marginal would be calculated using

$$p(x_2) = \iiiint p(x_1, x_2, x_3, x_4, x_5)\, \mathrm{d}\boldsymbol{x}_{\setminus 2}. \tag{17}$$

However, as the model can be factorized, we can rewrite this computation into smaller and more efficient steps. We plug in the factorized form (16) into (17) and apply the distributive law of integration. This rewriting of the marginalization leads to

$$p(x_2) = \underbrace{\int f_a(x_1, x_2)\, \mathrm{d}x_1}_{\overrightarrow{\mu}_{x_2}} \cdot \underbrace{\iint f_b(x_2, x_3, x_4) \underbrace{f_c(x_3)}_{\overleftarrow{\mu}_{x_3}} \underbrace{\left(\int f_d(x_4, x_5)\, \mathrm{d}x_5\right)}_{\overleftarrow{\mu}_{x_4}} \mathrm{d}x_3 \mathrm{d}x_4}_{\overleftarrow{\mu}_{x_2}} \tag{18}$$

The global execution of (17) is now reduced to a set of lower-dimensional integrals, which can be interpreted as messages. These messages are denoted by μ and can be computed locally by the nodes. Figure 5 shows the messages given in (18) in the FFG. In theory, FFGs are undirected, but in order to distinguish forward and backward messages on the edges, we have added arrows. The messages can be thought of as a summary from the corresponding dashed box. By passing on the messages between the nodes through the graph, inference is realized.

This example illustrates the idea of the sum-product rule. The sum-product rule states that for an arbitrary given node $f(x_1, x_2, \ldots, x_n)$ with messages $\overrightarrow{\mu}_{x_{\setminus j}}$ flowing in, the message $\overrightarrow{\mu}_{x_j}$ flowing out is given by

$$\overrightarrow{\mu}_{x_j} = \int f(x_1, x_2, \cdots, x_n) \prod_{i \neq j} \overrightarrow{\mu}_{x_i} \mathrm{d}x_{\setminus j}. \tag{19}$$

This sum-product rule is the core of the sum-product message passing algorithm, also known as belief propagation. A more detailed explanation of sum-product message passing in FFGs can be found in [17].

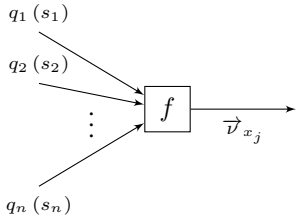


Figure 6. Situational sketch of variational message passing at a factor node $f(s_1, s_2, \ldots, s_n)$ with incoming messages $q(x_{\setminus j})$ and outgoing variational message $\overrightarrow{\nu}_{x_j}$.

C. Variational message passing

Sum-product message passing is an exact inference algorithm and can only be used in the case of tractable computations. However, non-conjugate relationships between variables can lead to intractable computations. Due to our choice to model the sound profile mean with a Gaussian distribution and the sound profile precision matrix with a Wishart distribution, we have such a non-conjugate relationship. In such cases, we can resort to an approximate message passing algorithm, called variational message passing (VMP) [18], [19].

Suppose that we have a generative model $p(\boldsymbol{s}, x)$, in which the computation of the posterior distribution $p(\boldsymbol{s} \mid x)$ involves some intractable computations. We therefore cannot compute the posterior distribution $p(\boldsymbol{s} \mid x)$ exactly. VMP introduces an approximate posterior distribution $q(\boldsymbol{s})$, which approximates the exact posterior and facilitates tractable computations. This approximate posterior distribution is obtained through the minimization of the variational free energy, which we've already introduced in (4) in Section II.

Considering an arbitrary node $f(s_1, s_2, \ldots, s_n)$, it can be shown [13], [18], [19] that minimization of the variational free energy is achieved by sending variational messages $\overrightarrow{\nu}$ of the form

$$\overrightarrow{\nu}_{x_j} \propto \exp\left(\int \prod_{i \neq j} q_i(s_i) \log f(s_1, s_2, \ldots, s_n) \mathrm{d}\boldsymbol{s}_{\setminus j}\right). \tag{20}$$

Figure 6 shows a situational sketch of the node. Using these messages, the approximate marginal $q_j(s_j)$ can be obtained by multiplying the incoming and outgoing messages on the respective edge s_j as

$$q_j(s_j) = \overrightarrow{\nu}_{s_j} \cdot \overleftarrow{\nu}_{s_j}. \tag{21}$$

V. Experimental validation

Now that we have introduced the solution framework and specified our models for road event detection specifically, we turn to evaluating the proposed solution framework by means of real data. As introduced in Section I, we use the scenario of

event detection along the road. The goal is to detect an event, while not marking environmental sounds and passing cars as events.

In this section, we will first describe the data used for this experiment. Then, we discuss the details of our implementation, touching upon data preprocessing, the message passing realization, and model variable choices. Finally, the results of the experiment are presented.

A. Data

The data that is used in this paper is twofold. First off, we use audio recordings of cars passing by. These recordings have been made using Sorama's Listener64 at the main road N737 near Enschede, the Netherlands. The recordings have been sampled at 32.5 kHz, and have a duration of 6 seconds each. Next to these recordings of passing cars, we use a selection of the MIVIA audio events data set [9], [20]. This data set is composed of a total of 400 events for road surveillance applications, containing tire skidding and car crashes, superimposed to a typical road background sound. The sounds have been registered using an Axis T83 omnidirectional microphone for audio surveillance applications, sampled at 32 kHz.

B. Implementation details

1) Data preprocessing: The data is first resampled to 16 kHz, as little information is present in the spectrum above 8 kHz. Next, an audio fragment is generated by concatenating sounds of passing cars and sounds of events.

For our experiment, we have concatenated two different fragments of a passing car (dubbed *car1* and *car2*, taken from the Sorama data set), a fragment of yet another car approaching followed by this car crashing (dubbed *car3 + crash*, a custom-made combination of two single fragments taken from the Sorama data set and the MIVIA data set), and a slipping car (dubbed *slip*, taken from MIVIA data set), in that respective order. This results in a total length of about 25 seconds. The choice of the fragments of passing cars was made with the gradual behavior in mind, aiming to mimic a realistic time signal, without too abrupt changes in sound between consecutive car passing fragments. The concatenation of the separate segments is illustrated in Figure 7.

As indicated in Section III, we assume log-power spectrum coefficients as input for our models. Hence, we need some preprocessing to transform the concatenated time signal of our data to the log-power spectrum domain. Consider the resampled, concatenated, and real-valued discrete-time signal $\boldsymbol{y} = [y_1, y_2, \ldots, y_L]^\top$ of length L. In order to compute the log-power spectrum, we use signal segments of length $Q = 160$, corresponding to segments of 10 milliseconds. To each segment we apply a Hanning window, and we use an overlap of 50%, resulting in $N = 2\frac{L}{Q} - 1$ segments and a time resolution of 5 ms. For each of the N signal segments, we compute the discrete Short-time Fourier transform (STFT), resulting in Q complex frequency coefficients per segment. As we are dealing with a real-valued signal, the frequency spectrum is symmetrical around DC. Therefore, we omit the redundant half the coefficients, resulting in a vector of

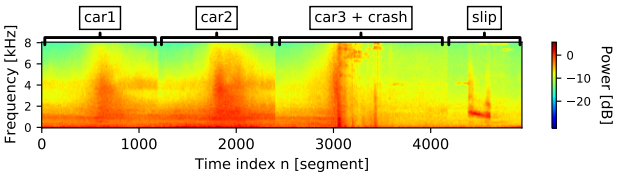


Figure 7. This figure shows a spectrogram representation of the input signal $\boldsymbol{x}_n$ for the experiment, as described in Section V-B1. The annotations on top indicate the original separate segments for reference.

$M = \lfloor \frac{Q}{2} \rfloor + 1$ frequency coefficients for each signal segment $n \in \{1, \ldots, N\}$. In the last steps, we consecutively take the absolute value, square, and take the natural logarithm of each of the coefficients in this vector to end up with the desired log-power spectrum coefficients vector $\boldsymbol{x}_n$ for each signal segment n. This signal acts as observations for both our models and is shown as a spectrogram in Figure 7, along with annotations of the original separate fragments.

2) Inference: The computations in this research are done using the scientific programming language Julia [21]. The ForneyLab package [22] is used to perform the inference tasks as introduced in Section II. This package automatically generates message passing-based inference algorithms for probabilistic models specified by the user, which allows for swift model iterations.

In ForneyLab, parameters θ are treated as an "extended" state, hence they are also updated during state estimation. In order to accomplish our parameter estimation stage, we update the parameters $\boldsymbol{\theta}$ for the duration of the segment *car1*, which equals 6 seconds, while updating the posterior state distribution. After that, we no longer update the parameters. Because of this, we also have the variational free energy and posterior state updates available for the parameter estimation stage.

3) Model and framework variables: For running the experiment, we have set values for all the variables that are present in our framework and defined models. We use equal model probabilities $p(m_1) = p(m_2) = 0.5$ as a starting point, as we mainly focus on the validation of the framework and models in general. The observation noise is kept the same for m_1 and m_2 via an identical Λ_x. By using a diagonal precision matrix, we assume i.i.d. noise.

The mean $\boldsymbol{\mu}$ of $\boldsymbol{m_h}$ is set on the values of the first observation, as this is assumed already close to the final values of the sound profile. For the degrees of freedom ν of the prior of the Wishart, we have based ourselves on the degrees of freedom initiated by the vague Wishart prior in ForneyLab, equaling the dimension of the state, M. As the mean of a Wishart distribution equals $\nu \boldsymbol{V}$, we chose $\boldsymbol{V} = \nu^{-1}\boldsymbol{I}$ as prior, which means that the initial precision matrix for the sound profile $\boldsymbol{h}$ equals $\boldsymbol{I}$. For the mean of the state of the baseline model $\boldsymbol{\mu_z}$, we use the mean value of each frequency bin for the duration of the *car1* segment, and low values in precision matrix $\Lambda_{\boldsymbol{z}}$ to ensure broad coverage, leading to an average, mediocre model behavior that can serve as a baseline.

The scaling of each identity matrix $\boldsymbol{I}$ for the involved the

Table I
VALUES OF PRIORS AND OTHER VARIABLES IN THE MODELS AND FRAMEWORK USED FOR THE CONDUCTED EXPERIMENT

Variable	Value	Equation
$p(m_1)$	0.5	(6)
$p(m_2)$	$1 - p(m_1)$	(6)
Λ_x	$(1 \cdot 10^6) \cdot \boldsymbol{I}$	(8), (13)
μ_{g_0}	0	(12)
γ_{g_0}	$1 \cdot 10^{-2}$	(12)
$\boldsymbol{\mu}$	$\boldsymbol{x}_1$	(10a)
Λ	$(1 \cdot 10^{-10}) \cdot \boldsymbol{I}$	(10a)
ν	M	(10b)
$\boldsymbol{V}$	$\nu^{-1} \cdot \boldsymbol{I}$	(10b)
γ_g	$1 \cdot 10^2$	(11)
$\boldsymbol{\mu_z}$	$\texttt{mean}(\boldsymbol{x}_{1:1200})$	(14)
$\Lambda_{\boldsymbol{z}}$	$(2 \cdot 10^{-2}) \cdot \boldsymbol{I}$	(14)

precision matrices, and all the other values not specifically mentioned before have been determined by means of trial and error. Table I shows an overview of all the variable values used in our experiment, along with a reference to the equation they belong to.

C. Results

Figure 8 shows the results of our solution framework and model specification, based on the input signal $\boldsymbol{x}_n$, defined in the previous subsection. More specifically, the second plot shows the estimated state of the main model over time, and the third plot the accompanying variational free energy in the main and baseline model. The fourth plot in Figure 8 shows the posterior model probability for the baseline model, representing the event detection result. The fifth plot shows a reconstruction of the input signal, based on the inferred parameters and tracked state. Figure 9 shows a simplified distribution of the inferred sound profile $\boldsymbol{h}$. For plotting convenience, we use $\boldsymbol{\mu}$ as a mean value, represented by the bold blue line, and a simplified mean $\nu \cdot \texttt{diag}(\boldsymbol{W_h})$ of Wishart distribution for calculation of the standard deviation. The shaded light blue covers the range up to one standard deviation difference to the mean.

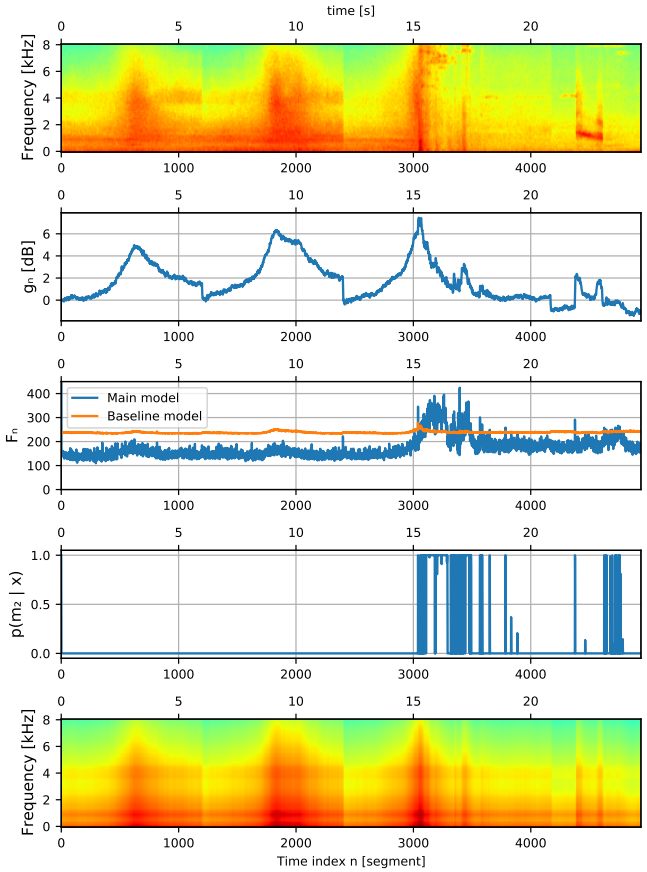


Figure 8. This figure again presents the input signal (for reference), plus an overview of the results of our presented solution framework based on this input data. The top plot shows this input signal $\boldsymbol{x}_n$ as a spectrogram. The second plot shows the estimated state of the model (gain) over time, acting on this input signal. The third plot shows the variational free energy over time of both our models. The fourth plot shows the posterior model probability of the baseline model m_2 over time, representing the event detection. The fifth plot shows the reconstructed input signal based on the inferred parameters and tracked gain. The label for the upper x-axis of the top plot and the label n on the lower x-axis of the bottom plot count for all the plots.

VI. RELATED WORK

In this section, we present an overview of related work. First, we take a glance at the available literature on generative acoustic modeling. Next, an overview of acoustic event detection work is given.

An overview of approaches to sound modeling is given by [23]. "Physical modeling" mathematically models the physics that generate the target waveform. Consequently, the parameters are of a physical nature, making them intuitive to work with. An example of physical modeling can be found in [24]. We also apply physical modeling in our main model. The "acoustic modeling" approach uses signal generators and processors for manipulating waveforms just like a synthesizer, such as oscillators, modulators, and filters. However, the parameters are more algorithm-specific, making them not as intuitive for control as parameters in physical modeling.

In the 1980s, the Hidden Markov Model (HMM) marked the eminence of statistical models, dominating the automatic

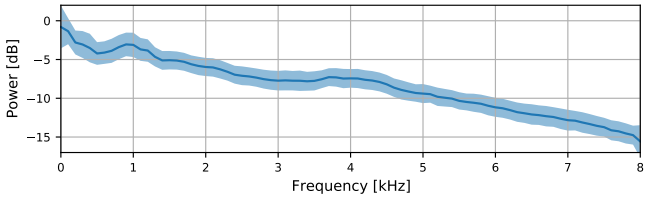


Figure 9. This figure shows the (simplified) distribution of sound profile $\boldsymbol{h}$, as a result of the parameter estimation stage. The solid line corresponds to a mean, and the shadowed region corresponds to one standard deviation below and above this mean. This sound profile is inferred during the parameter estimation stage. For plotting convenience, we use $\boldsymbol{\mu}$ as a mean, and a simplified mean $\nu \cdot \texttt{diag}(\boldsymbol{W_h})$ of Wishart distribution for calculation of the standard deviation (see equation set (10) for meaning of parameters).

generation (and recognition) of speech. These models learn their parameters from data, contrary to the man-made design in the case of physical and acoustic modeling. Therefore, this was the start of data-driven statistical parametric models.

The modern successors of HMMs are deep generative models. Generative neural modeling of audio has been explored for some time, albeit mainly in other audio representations than waveform, like spectrograms [25]. However, generative models capable to generate audio waveforms directly in the time domain were explored more recently. The latter was long considered to be infeasible due to the scale of the problem as a result of the sample rates surpassing at least 16 kHz [26]. Examples of those explorations include WaveNet [27], VRNN [28], WaveRNN [29] and SampleRNN [30]. These models predict digital waveforms one timestep at a time and rely on autoregressive (AR) models. WaveNet is a convolutional neural network (CNN) that includes dilated convolutions [31]. VRNN and WaveRNN are recurring neural networks (RNNs). SampleRNN uses RNNs as well but it combines RNNs running at different clock rates to model longer term dependencies in audio waveforms. They all model the distribution as a product of conditionals: $p(\boldsymbol{x}) = \prod_t p(x_t|x_{<t})$ with $t = 1, ..., T$ [26]. Recently, there are also attempts to use Generative Adversarial Networks (GANs) as an alternative approach for the generation of audio waveforms, like WaveGAN [32].

An interesting division between generative neural networks is given by [23]. They claim that many generative models are trained using maximum likelihood, and therefore, they can be classified in how they represent the likelihood. On the one hand, there are networks that facilitate an explicit formulation of the marginal likelihood, like the mentioned AR models, or variational autoencoders (VAEs). For those models, by means of the explicit likelihood, a measure is provided for determining how close the model is to the real data. On the other hand, there are the models that only have implicit knowledge of probability distributions, and therefore do not have an unambiguous way to evaluate the model, like GANs. Within the group with explicit probability densities, a subdivision can be made between generative neural models where those densities are tractable (AR models), and models where there are approximations involved for the explicit density (VAEs).

Acoustic event detection is not a new research field; various research efforts have been conducted in this field. Early works based themselves on extracting feature sets from the input audio signal, based on concepts like log-frequency banks, Mel-frequency Cepstral Coefficients (MFCC) [33], but also features based on temporal and spectral characteristics of the signal. Detection and recognition tasks are performed on the selected feature sets using classifiers, such as Gaussian Mixture Model-based (GMM) classifiers [34] or support vector machines (SVMs) [33], [35]

Where in the beginning the features and representations were handcrafted, later on, research focused on a more data-driven approach by learning representations from actual training data, e.g. based on the bag-of-words approach [36], [37], [38]. In [9], for example, the focus is on road surveillance, more specifically on automatic detection of tire skidding and car crashes. They extract low-level feature vectors from the audio stream. Based on a data-based audio dictionary, the nearest audio word is determined. Over a certain time interval, histograms of these audio words are computed, which are then classified using SVMs, resulting in a final class. For experimental validation, they created a publicly available MIVIA road event dataset, on which they achieved an average accuracy of 78.95%.

Recently, also deep learning approaches were proposed to learn effective representations of sound. [39] applies two RNNs to detect and classify an event, respectively. [8] takes a deep learning approach in the form of a Convolutional Neural Network (CNN). More specifically, it inherits the MobileNet network [40], [41], which proves to be particularly fitted for real-time application in embedded systems due to its good trade-off between accuracy, complexity in terms of floating-point operations (FLOPs), and the number of parameters [42]. It achieves an event recognition rate of over 99% on the previously mentioned MIVIA road events data set.

VII. Discussion

In this section, we discuss the results of Section V and some of the limitations of this research. Furthermore, we present directions for future work.

In Figure 8 we see that the gain of the main model follows the signal in a gradual way most of the time for the signal parts *car1*, *car2*, and (the first part of) *car3 + crash*. This is what we would expect because this represents the gradual motion of a car; increasing when the car approaches, and decreasing when the car has passed by. The fact that this is the case for *car1* already (so during the parameter estimation stage) indicates that the model grasps this fairly quickly. The sudden drops in the gain around $n = 1200$, $n = 2400$, and $n = 4200$ can be explained by the non-smooth concatenation of different signal fragments. The gain introduces less-fluent behavior (sudden drops and rises) when the events are introduced, like around $n = 3400$ and $n = 4200$.

If we look at the plot of the simplified inferred sound profile $\boldsymbol{h}$ in Figure 9, we can see that the standard deviation is slightly higher for the frequencies up to 1.5 kHz compared to the rest of the spectrum. Looking at the plot of the input signal $\boldsymbol{x}_n$, specifically for the *car* signal fragments, we can see that while for the higher frequencies there is a gradual increase and decrease in power over time, this is not the case for the frequencies up to 1.5 kHz. Over time they seem to remain relatively constant; the gain has less to no impact here. As a result, during parameter estimation, these frequencies end up with a larger uncertainty, hence a larger standard deviation.

With our choice for the main model, we have taken a physical modeling approach to model the characteristic behavior, with the model existing out of comprehensible components related to a physical nature. The sound profile catches the acoustic blueprint of a car, while the gain captures the motion relative to the sensor. Looking at both the gain and the sound profile, the main model seems to catch what we designed it for. If we look at the variational free energy of the main model in Figure 8, it shows a rather stable behavior during the expected behavior once the parameter estimation stage has

been completed. This shows that the main model works as intended.

Besides this observation, two other observations can be made. First of all, a notable slight increase in variational free energy can be observed at around $n = 1850$, when the gain reaches its local maximum. We relate that to the non-scaling power in the lower frequencies up to 1.5 kHz, also touched upon in the previous paragraph. This can also be noted in the reconstructed signal on the bottom plot of Figure 8, where there is more power than for the original signal around this time. Secondly, we see a peak at around $n = 2400$, which can be explained by the concatenation of the signal, just like the drop in gain, as mentioned before. Once the events pop up in the signal, the variational free energy rises considerably, as expected, as this is concerns unexpected deviation from expected behavior, which the model cannot represent well.

The variational free energy of the baseline model is relatively constant over time, during the cars passing as well as during the events. This is exactly what we designed the model for. Slight increases can be noted at times where the gain is also high. These increases can be explained by the fact that the input signal reaches peak values that differ relatively a lot from the actual prediction of the model. This results in lower evidence, hence higher variational free energy.

The actual event detection result, shown in the bottom plot of Figure 8, shows a rather binary result. This can be explained by the fact that the approximation for the posterior model probability of m_2 (7) resembles a sigmoid function $S(x) = \frac{1}{e^{-x}+1}$ where $x = -\left(F_1 - F_2 - \log \frac{p(m_1)}{p(m_2)}\right)$. The active region of a sigmoid function ranges approximately from $x = -5$ to $x = 5$ and is close to either 0 or 1 outside this domain. As our x falls out of this range most of the time, this results in a rather binary behavior. In that sense, the prior model probabilities only influence the result when the difference in variational free energies nears the active domain of the sigmoid function. However, it should be noted that this relationship in not exact, as variational free energy bounds the negative log-evidence and only serves an approximation.

Regarding the detection of the actual events, it is striking that only the start of the slipping is detected. If we look at the reconstructed signal in the bottom plot of Figure 8, we see that the high-energy slipping part falls in approximately the same range as the high-energy frequencies of the sound profile. Looking at the reconstruction and the gain, the model sees it as part of the sound profile. The event detection triggers after the slip, at round $n = 6000$, are caused by a high-frequency sound that is part of the background noise of this data.

Some comments have to be made concerning this research. First, our model takes all sounds into consideration for the sound profile during parameter estimation, despite maybe not being related to the vehicle in motion. Think of background sounds like weather conditions. Furthermore, the model does not support (relative) silence. Also, the experiment we have conducted is not ideal. The car sounds have been recorded at a main road without post-processing, but the event sounds of the MIVIA data set are an artificial combination of an acoustic road event and non-original background noise. Therefore, the input signal is not an ideal representation of a real-life scenario. Furthermore, exact performance is yet to be determined. Moreover, we have validated our framework on only a single test. Despite the promising result, further testing is required to substantially validate our framework and models.

Due to the limitations of this research – as a result of the choices made – further research is invited. This research should consider further iterations on model design, for example by including the Doppler effect as a physical modeling component. Also, further research should look into incorporating background noise separately from the sound profile. This might make lower-frequency events like car slipping easier to detect. It might also be worth considering excluding the lower frequencies from the model while taking into account slipping frequency. Besides including background noise, it should be considered to exclude a car sound profile in case of silence. Furthermore, our model is based on log-power spectrum coefficients, which require preprocessing of the time signal. It would be interesting to investigate whether improvements in performance or success rate can be made here by incorporating this preprocessing into the model. With regard to the baseline model, we chose a substantially stripped, non-updating model in this research, without any parameter or state estimation involved. Despite the baseline model does its job, the variable choices could be improved. Further research could look into using slightly more advanced approaches for determining the state distribution parameters, allowing for parameter estimation, for example.

That fact that we can detect acoustic road events can be put into practice in multiple ways. First of all, the detection can act as a direct trigger for emergency team dispatch. This will save time and decrease mortality rate, as seen in Section I. Detected events could also serve as a prioritizer of video streams in a road monitoring control room. This both serves a curing role. Detected events can also be of less-hazardous but implicating behavior, like tire slipping. This information can be used to improve safety and therefore serve a preventional purpose. Either way, society can benefit from it.

While in this paper we focus on a single application, our methodology allows us to perform event detection in any context and is not restricted to a specific type of problem. For other applications, such as proactive maintenance on factory floors or crowd control, we only need to iterate the model specification until we have obtained an effective model.

VIII. Conclusion

In this paper, we have presented a general Bayesian framework for situated detection of acoustic road events using probabilistic generative models. We have described the stages in this framework, i.e., parameter estimation, state estimation, and model comparison, as probabilistic inference tasks that can be executed automatically by means of message passing. Moreover, we demonstrated the validity of this framework with an experiment on a road event detection application. We defined the required models to match this application and applied them to an artificial application environment. We were able to detect acoustic events in traffic such as a crash and tires slipping using our introduced framework and defined models.

REFERENCES

[1] T. Gandhi and M. M. Trivedi, "Pedestrian Protection Systems: Issues, Survey, and Challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, p. 18, 2007.

[2] S. Rauscher, G. Messner, P. Baur, J. Augenstein, K. Digges, E. Perdeck, G. Bahouth, and O. Pieske, "Enhanced automatic collicsion notification system – Improved rescue care due to injury prediction – First field experience," *The 21st International Technical Conference on the Enhanced Safety of Vehicles Conference (ESV)-International Congress Center Stuttgart, Germany*, pp. 09–0049, 2009.

[3] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt, "WreckWatch: Automatic Traffic Accident Detection and Notification with Smartphones," *Mobile Networks and Applications*, vol. 16, no. 3, pp. 285–303, Jun. 2011. [Online]. Available: http://link.springer.com/10.1007/s11036-011-0304-8

[4] M. S. Shirazi and B. T. Morris, "Looking at Intersections: A Survey of Intersection Monitoring, Behavior and Safety Analysis of Recent Studies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 4–24, Jan. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7556973/

[5] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic Flow Prediction With Big Data: A Deep Learning Approach," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–9, 2014. [Online]. Available: https://ieeexplore.ieee.org/document/6894591

[6] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic Flow Prediction for Road Transportation Networks With Limited Traffic Data," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2014. [Online]. Available: https://ieeexplore.ieee.org/document/6878453

[7] M. Cristani, M. Bicego, and V. Murino, "Audio-Visual Event Recognition in Surveillance Video Sequences," *IEEE Transactions on Multimedia*, vol. 9, no. 2, p. 11, 2007.

[8] P. Foggia, A. Saggese, N. Strisciuglio, M. Vento, and V. Vigilante, "Detecting sounds of interest in roads with deep networks," in *Image Analysis and Processing – ICIAP 2019: 20th International Conference*. Springer Verlag, Sep. 2019, pp. 583–592. [Online]. Available: https://research.utwente.nl/en/publications/detecting-sounds-of-interest-in-roads-with-deep-networks

[9] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Audio Surveillance of Roads: A System for Detecting Anomalous Sounds," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 279–288, Jan. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7321013/

[10] E. Babaee, N. B. Anuar, A. W. Abdul Wahab, S. Shamshirband, and A. T. Chronopoulos, "An Overview of Audio Event Detection Methods from Feature Extraction to Classification," *Applied Artificial Intelligence*, vol. 31, no. 9-10, pp. 661–714, Nov. 2017. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/08839514.2018.1430469

[11] M. Crocco, M. Cristani, A. Trucco, and V. Murino, "Audio Surveillance: A Systematic Review," *ACM Computing Surveys*, vol. 48, no. 4, pp. 1–46, May 2016. [Online]. Available: https://dl.acm.org/doi/10.1145/2871183

[12] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, May 2015. [Online]. Available: http://www.nature.com/articles/nature14541

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006. [Online]. Available: http://www.springer.com/computer/image+processing/book/978-0-387-31073-2

[14] S. Särkkä, *Bayesian Filtering and Smoothing*. London ; New York: Cambridge University Press, Oct. 2013.

[15] H. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The Factor Graph Approach to Model-Based Signal Processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, Jun. 2007, conference Name: Proceedings of the IEEE.

[16] G. D. Forney, "Codes on graphs: normal realizations," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001, conference Name: IEEE Transactions on Information Theory.

[17] S. Korl, "A factor graph approach to signal modelling, system identification and filtering," Ph.D. dissertation, ETH Zurich, 2005, artwork Size: 187 S. Medium: application/pdf Pages: 187 S. [Online]. Available: http://hdl.handle.net/20.500.11850/82737

[18] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational Inference: A Review for Statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, Apr. 2017, arXiv: 1601.00670. [Online]. Available: http://arxiv.org/abs/1601.00670

[19] J. Dauwels, "On Variational Message Passing on Factor Graphs," in *IEEE International Symposium on Information Theory*, Jun. 2007, pp. 2546–2550. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/4557602

[20] P. Foggia, A. Saggese, N. Strisciuglio, and M. Vento, "Cascade classifiers trained on gammatonegrams for reliably detecting audio events," in *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Seoul, South Korea: IEEE, Aug. 2014, pp. 50–55. [Online]. Available: https://ieeexplore.ieee.org/document/6918643

[21] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A Fresh Approach to Numerical Computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, Jan. 2017. [Online]. Available: https://epubs.siam.org/doi/10.1137/141000671

[22] M. Cox, T. van de Laar, and B. de Vries, "A factor graph approach to automated design of Bayesian signal processing algorithms," *International Journal of Approximate Reasoning*, vol. 104, pp. 185–204, Jan. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888613X18304298

[23] M. Huzaifah and L. Wyse, "Deep generative models for musical audio synthesis," *arXiv:2006.06426 [cs, eess, stat]*, Jun. 2020, arXiv: 2006.06426. [Online]. Available: http://arxiv.org/abs/2006.06426

[24] J. O. Smith, "Physical Modeling Using Digital Waveguides," *Computer Music Journal*, vol. 16, no. 4, pp. 74–91, 1992, publisher: The MIT Press. [Online]. Available: https://www.jstor.org/stable/3680470

[25] S. Vasquez and M. Lewis, "MelNet: A Generative Model for Audio in the Frequency Domain," *arXiv:1906.01083 [cs, eess, stat]*, Jun. 2019, arXiv: 1906.01083. [Online]. Available: http://arxiv.org/abs/1906.01083

[26] S. Dieleman, A. v. d. Oord, and K. Simonyan, "The challenge of realistic music generation: modelling raw audio at scale," *arXiv:1806.10474 [cs, eess, stat]*, Jun. 2018, arXiv: 1806.10474. [Online]. Available: http://arxiv.org/abs/1806.10474

[27] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *arXiv:1609.03499 [cs]*, Sep. 2016, arXiv: 1609.03499. [Online]. Available: http://arxiv.org/abs/1609.03499

[28] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio, "A Recurrent Latent Variable Model for Sequential Data," *arXiv:1506.02216 [cs]*, Apr. 2016, arXiv: 1506.02216. [Online]. Available: http://arxiv.org/abs/1506.02216

[29] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. v. d. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient Neural Audio Synthesis," *arXiv:1802.08435 [cs, eess]*, Jun. 2018, arXiv: 1802.08435. [Online]. Available: http://arxiv.org/abs/1802.08435

[30] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An Unconditional End-to-End Neural Audio Generation Model," *arXiv:1612.07837 [cs]*, Feb. 2017, arXiv: 1612.07837. [Online]. Available: http://arxiv.org/abs/1612.07837

[31] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," *arXiv:1511.07122 [cs]*, Apr. 2016, arXiv: 1511.07122. [Online]. Available: http://arxiv.org/abs/1511.07122

[32] C. Donahue, J. McAuley, and M. Puckette, "Adversarial Audio Synthesis," *arXiv:1802.04208 [cs]*, Feb. 2019, arXiv: 1802.04208. [Online]. Available: http://arxiv.org/abs/1802.04208

[33] Guodong Guo and S. Z. Li, "Content-based audio classification and retrieval by support vector machines," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 209–215, Jan. 2003, conference Name: IEEE Transactions on Neural Networks.

[34] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli, "Audio Based Event Detection for Multimedia Surveillance," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 5, May 2006, pp. V–V, iSSN: 2379-190X.

[35] A. Rabaoui, M. Davy, S. Rossignol, and N. Ellouze, "Using One-Class SVMs and Wavelets for Audio Surveillance," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 763–775, Dec. 2008, conference Name: IEEE Transactions on Information Forensics and Security.

[36] V. Carletti, P. Foggia, G. Percannella, A. Saggese, N. Strisciuglio, and M. Vento, "Audio surveillance using a bag of aural words classifier," in *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, Aug. 2013, pp. 81–86.

[37] J.-J. Aucouturier, B. Defreville, and F. Pachet, "The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music," *The Journal of the*

Acoustical Society of America, vol. 122, no. 2, pp. 881–891, Aug. 2007. [Online]. Available: http://asa.scitation.org/doi/10.1121/1.2750160

[38] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Reliable detection of audio events in highly noisy environments," *Pattern Recognition Letters*, vol. 65, pp. 22–28, Nov. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167865515001981

[39] F. Colangelo, F. Battisti, M. Carli, A. Neri, and F. Calabró, "Enhancing audio surveillance with hierarchical recurrent neural networks," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug. 2017, pp. 1–6.

[40] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv:1704.04861 [cs]*, Apr. 2017, arXiv: 1704.04861. [Online]. Available: http://arxiv.org/abs/1704.04861

[41] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *arXiv:1801.04381 [cs]*, Mar. 2019, arXiv: 1801.04381. [Online]. Available: http://arxiv.org/abs/1801.04381

[42] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark Analysis of Representative Deep Neural Network Architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018, conference Name: IEEE Access.