Variational autoencoder for the identification of piecewise models ⋆

Manas Mejari Marco Forgione Dario Piga

IDSIA Dalle Molle Institute for Artificial Intelligence, USI-SUPSI, Via la Santa 1, CH-6962 Lugano-Viganello, Switzerland. (e-mail: {manas.mejari, marco.forgione, dario.piga}@supsi.ch).

Abstract: The paper presents a *variational autoencoder* (VAE) tailored for the identification of hybrid *piecewise* models in input-output form. We show that using a specialized autoencoder structure, the *latent space* can provide an interpretable representation in terms of the modes of the underlying hybrid system. In particular, we use categorical encoding of the discrete latent variables whose distribution is approximated via the *encoder* neural network, characterizing a partition of the regressor space, while the *decoder* consists of a set of neural networks, each corresponding to a local submodel of the piecewise hybrid system. By employing variational Bayesian framework for inference, the constitutive terms of the *evidence lower bound* (ELBO) are derived analytically with the chosen VAE architecture. The ELBO loss consists of a reconstruction error term and a regularization term over the latent modes. This loss is optimized in order to train the encoder-decoder networks concurrently via back-propagation. The developed framework is not restricted to simple *piecewise affine* (PWA) models and it can be straightforwardly extended to general class of piecewise non-linear systems over non-polyhedral domains.

Keywords: System identification; hybrid systems; PWA systems; Variational autoencoders.

1. INTRODUCTION

The availability of flexible software with automatic differentiation capabilities along with the hardware support for parallelization has led to the development of algorithms using neural networks for the identification of *non-linear* dynamical systems (Ljung et al., 2020; Forgione and Piga, 2020; Masti and Bemporad, 2021; Piga et al., 2021; Wang, 2017). In comparison, relatively less attention is given for developing a deep learning framework for modeling *hybrid* systems, which operate concurrently in continuous as well as discrete domains. Hybrid models are a powerful tool to describe the behavior of many real-world systems which are characterized by different operating regions or modes.

Most of the conventional approaches for hybrid system identification (see, survey paper (Garulli et al., 2012) for an overview) are restricted to simple hybrid structures (such as piecewise affine maps), often requiring *linear* separability assumptions of the data clusters. These methods do not utilize powerful deep learning tools, which can potentially be exploited to recognize *non-linear* clustering patterns and more complex local dynamics. In this paper, we aim at developing a framework for learning a class of hybrid dynamical models called *piecewise models*, utilizing tools from modern deep learning.

A piecewise model consists of a set of local submodels, each defined over a specific region of the regressor space. Depending upon the parameterization of local submodels as well as its corresponding region, the piecewise models are classified into *piecewise affine* (PWA), *piecewise non-linear* (PWN), *non-linearly piecewise affine* (NPWA) and *nonlinearly piecewise nonlinear* (NPWN) model classes (Lauer and Bloch, 2008). Learning piecewise models from data is an NP-hard problem (Lauer, 2015), which requires estimating the sub-models as well as a partition of the regressor space. Over the years several heuristics have been developed for learning piecewise models (albeit, not utilizing deep learning), which include: the recently proposed optimization-based algorithm (Bemporad, 2022) for handling numeric as well as categorical data; the bounded-error approach (Bemporad et al., 2005); recursive clustering-based methods (Boukharouba et al., 2009; Breschi et al., 2016; Mejari et al., 2020a), mixed-integer programming algorithms (Roll et al., 2004; Mejari et al., 2020b), Bayesian inference (Juloski et al., 2005; Piga et al., 2020), among many others. The aforementioned contributions and most of the existing approaches proposed in the literature are restricted to PWA models having *affine* submodel parameterization over *polyhedral* partitioning. Very few works have addressed the identification of non-linear submodels and non-polyhedral domains, *e.g.,* in (Lauer and Bloch, 2008), kernel regression and *support vector machines* are employed to identify PWN and NPWA models. Kernel methods (for submodel estimation) combined with optimization-based strategies (for clustering) are proposed in (Lauer and Bloch, 2014; Mazzoleni et al., 2021) to learn PWN models. In this paper, we propose a method based on *variational autoencoder* (VAE) to learn piecewise models. The VAE was first introduced in the seminal work (Kingma and Welling, 2014), generalizing the

⋆ This work has been supported by HASLER STIFTUNG under the project *INHALE: Interpretable Neural networks for Hybrid dynamicAL systEms.*

deterministic autoencoderto the families of probabilistic models with variational Bayesian inference. In our work, we consider specific architectures for the *encoder* and *decoder* networks in VAE, in order to take into account the *hybrid* nature of the underlying system. The encoder approximates a distribution of the discrete latent modes, characterizing a partition of the regressor space, while the decoder consists of a set of neural networks, each corresponding to a local submodel of the piecewise hybrid system. With categorical encoding of the latent modes, we compute the expectation terms in the ELBO loss analytically. This avoids sampling and re-parameterization trick employed in the conventional VAE (Kingma and Welling, 2014), reducing the overall training cost.

To the best of our knowledge, neural networks have been considered for piecewise models only in (Brusaferri et al., 2020), where a neural network classifier to recognize the partition of the regressor space is proposed for identification of NPWA models. Our work differs from (Brusaferri et al., 2020) in the following ways: (i) in (Brusaferri et al., 2020), an expectation-maximization algorithm is employed in a *frequentists* setting to derive maximum-likelihood (ML) estimate of the model parameters. Our method is based on amortized variational inference in a *Bayesian* framework, which allows incorporating prior information of the latent mode probabilities via a *prior* distribution; (ii) the method in (Brusaferri et al., 2020) is restricted to *affine* parameterization of the submodels (PWA and NPWA models), while our work is applicable to non-linear submodels using decoder neural networks, *i.e.*, identification of more general classes of piecewise models such as PWN and NPWN; (iii) the ELBO loss derived in this paper consists of a regularization term in the form of mode entropy, which allows to control potential mode collapses during training as well as to impose regular structure over the latent space.

2. HYBRID PIECEWISE MODELS

In this section, we describe types of piecewise hybrid models which we aim to identify. We consider models in input-output form with inputs denoted as $x_t \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$, and measured outputs as $y_t \in \mathcal{Y} \subset \mathbb{R}^{n_y}$.

Piecewise affine model: The PWA model $f : \mathcal{X} \to \mathcal{Y}$ is described as follows:

$$f(x_t) = \begin{cases} \theta_1^{'} \left[\begin{smallmatrix} 1 \\ x_t \end{smallmatrix}\right] & \text{if } x_t \in \mathcal{X}_1, \\ \vdots & \vdots \\ \theta_K^{'} \left[\begin{smallmatrix} 1 \\ x_t \end{smallmatrix}\right] & \text{if } x_t \in \mathcal{X}_K, \end{cases} \tag{1}$$

where $K \in \mathbb{N}$ is the number of *modes* (*i.e.*, the number of affine functions defining f), and $\theta_i \in \mathbb{R}^{(n_x+1)\times n_y}$ is the parameter vector associated to the i-th affine submodel. The regions $\mathcal{X}_i \subseteq \mathcal{X}$ can be either polyhedral (PWA model) or non-polyhedral (NPWA model). The set $\{\mathcal{X}_i\}_{i=1}^{s}$ forms a *complete* partition [1] of the regressor space $\mathcal{X}$.

Probability weighted affine model: The *probability weighted affine* (PrA) model provides a smooth relaxation of the PWA model (Taguchi et al., 2009), where the individual models are composed by probabilistic weighting functions as follows:

$$y_t = f_{\text{pr}}(x_t) + v_t = \sum_{i=1}^{K} p_t^i \theta_i^{'} \left[\begin{smallmatrix} 1 \\ x_t \end{smallmatrix}\right] + v_t, \tag{2}$$

where p_t^i denotes the probability that the regressor x_t belongs to mode i and is parametrized by the softmax function $p_t^i = \frac{\exp(\eta_i^{'}\left[\begin{smallmatrix} 1 \\ x_t \end{smallmatrix}\right])}{1+\sum_{i=1}^{K-1}\exp(\eta_i^{'}\left[\begin{smallmatrix} 1 \\ x_t \end{smallmatrix}\right])}$ where $\eta_i \in \mathbb{R}^{n_x+1}$ is an unknown parameter vector characterizing the partition of the regresor space.

Piecewise non-linear model: Piecewise non-linear models represent a non-linear extension of PWA (1) ones, and are defined as follows:

$$f(x_t) = \begin{cases} g_1(x_t;\theta_i) & \text{if } x_t \in \mathcal{X}_1, \\ \vdots & \vdots \\ g_K(x_t;\theta_K) & \text{if } x_t \in \mathcal{X}_K, \end{cases} \tag{3}$$

where $g_i : \mathcal{X}_i \to \mathcal{Y}$ are local *non-linear* maps with parameters θ_i. As stated before, the regions $\mathcal{X}_i \subseteq \mathcal{X}$ can be either polyhedral (PWN model) or non-polyhedral (NPWN model).

3. LEARNING PROBLEM

We consider a training dataset $\mathcal{D} = \{x_t, y_t\}_{t=1}^{T}$ generated from the following system $\mathcal{S}$

$$y_t = f(x_t) + v_t, \tag{4}$$

where $v_t \sim \mathcal{N}(0, \sigma_v^2 I_{n_y})$ is a multivariate zero-mean white Gaussian noise with diagonal covariance, statistically independent of the input x_t.

Piecewise regression problem: Given a dataset $\mathcal{D}$, the piecewise regression problem entails the following tasks:

- **T1** Computation of the parameters $\Theta = [\theta_1, \cdots, \theta_K]$ defining the local affine functions in the PWA map f in (1); or equivalently f_{pr} in (2); or equivalently functions $\{g_i(x_t;\theta_i)\}_{i=1}^{K}$ in (3);
- **T2** Classification of the regressors x_t into clusters and subsequent characterization of the partition $\{\mathcal{X}_i\}_{i=1}^{K}$ of the regressor space.

In this work, we fix the number of modes K a-priori.

Active mode: In this paper, we will make use of a K dimensional discrete latent variable z_t, termed as the *active mode* at time t, having a 1-of-K representation, such that

$$z_t^i = \begin{cases} 1 & \text{if } x_t \in \mathcal{X}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Thus, the i-th component z_t^i of the vector z_t is 1 if and only if the regressor x_t belongs to the i-th region $\mathcal{X}_i$. The sequence of discrete active modes $\{z_t\}_{t=1}^{T}$, dictates the overall clustering of the regressor vectors $\{x_t\}_{t=1}^{T}$ to corresponding regions, which can be used to compute a partition of the regressor space $\mathcal{X}$.

Bayesian problem formulation: In the rest of the paper, we consider a *Bayesian* setting in which it is assumed that the output samples in $\mathcal{D}$ are generated from an (unknown) true distribution $y_t \sim p_\Theta(y_t|x_t)$ with parameters Θ of the piecewise hybrid system. The active mode z_t in

[1] The collection $\{\mathcal{X}_i\}_{i=1}^{K}$ is a complete partition of $\mathcal{X}$ if $\bigcup_{i=1}^{K} \mathcal{X}_i = \mathcal{X}$ and $\mathring{\mathcal{X}}_i \cap \mathring{\mathcal{X}}_j = \emptyset$, $\forall i \neq j$, with $\mathring{\mathcal{X}}_i$ denoting the interior of $\mathcal{X}_i$.

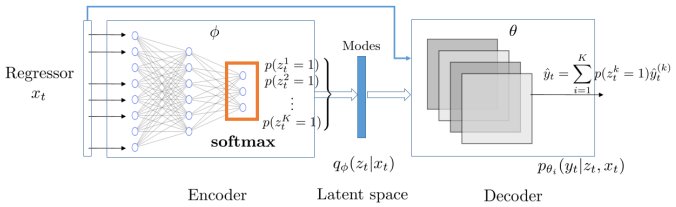

Fig. 1. Autoencoder architecture for PWA regression.

(5) represents latent encoding of the output y_t, which is assumed to be jointly distributed with y_t according to an unknown joint distribution $p_\Theta(z_t, y_t|x_t)$. In other words, the data generating system is modelled as $p_\Theta(y_t|x_t) = \sum_i^K p_\Theta(z_t^i, y_t|x_t) = \sum_i^K p_\Theta(y_t|z_t^i, , x_t)p_\Theta(z_t^i|x_t)$. Accordingly, the posterior over the latent modes is given by $p_\Theta(z_t|y_t, x_t) = \frac{p_\Theta(y_t|x_t,z_t)p_\Theta(z_t|x_t)}{p_\Theta(y_t|x_t)}$, where $p_\Theta(y_t|x_t, z_t)$ and $p_\Theta(z_t|x_t)$ denote the *likelihood* and the *prior*, respectively.

With this setting, the piecewise regression problem is formalized as follows:

Problem 1. We aim to estimate the parameters Θ of the piecewise model defining the data-generating distribution $p_\Theta(y_t|x_t)$ (corresponding to task **T1**) and to estimate the unknown posterior distribution $p_\Theta(z_t|y_t, x_t)$ over the latent modes, consequently identifying the mode sequence $\{z_t\}_{t=1}^N$ which characterizes a partition of the regressor space (i.e., task **T2**).

4. AMORTIZED VARIATIONAL INFERENCE

For inference of the piecewise models, we adopt the *variational autoencoder* (VAE) introduced in (Kingma and Welling, 2014). The VAE consists of an encoder neural network approximating the posterior over latent variables, followed by a decoder network which models the likelihood. The encoder-decoders networks are trained jointly by maximizing the *evidence lower bound* (ELBO) loss over the network parameters. In this work, we adapt and specialize the VAE concept for piecewise regression. In particular, the encoder characterizes a partition of the regressor space by recognizing active modes z_t, while the weights and biases of the decoder network represent the parameters Θ of the local submodels. In order to the compute posterior over the discrete latent modes z_t and to estimate the parameters Θ, we derive the corresponding ELBO loss for the piecewise models. By parameterizing encoder posterior with a categorical distribution, the expectation terms in the ELBO loss can be computed analytically, and thus, we avoid sampling and re-parameterization trick employed in the conventional VAE, reducing the computational effort during training.

We consider VAE network architecture shown in Fig. 1.

Encoder: The encoder $\mathcal{NN}_e(x_t; \phi)$ is a feed-forward neural network with trainable parameters ϕ and with the regressor x_t fed as input feature. The last layer of the encoder consists of a *softmax* activation function taken over K modes. Thus, the output of the encoder $\mu_t \in \mathbb{R}^K$ represents the probability of each mode at time t, given by

$$\mu_t = \mathcal{NN}_e(x_t; \phi), \tag{6}$$

where

$$\mu_t^i = p(z_t^i = 1), \quad \mu_t^i \in [0, 1], \quad i = 1, \ldots, K, \quad \sum_{i=1}^K \mu_t^i = 1$$

with z_t being the discrete latent variable characterizing the active mode as defined in (5).

Decoder: For PWA model, we define a decoder consisting of K independent neural networks, each corresponding to an affine submodel in the PWA map (1). Each of the K network consists of a single *linear* layer, weights (and biases) of which correspond to the parameters $\theta_i \in \mathbb{R}^{n_x}, i = 1, \ldots, K$ of the local affine function. The feature inputs to the i-th network are the regressor x_t and the probability of the i-th mode μ_t^i obtained from the encoder. The output of the i-th network is then given by $\hat{y}_t^i = \hat{\theta}_i' \begin{bmatrix} \mu_t^i \\ x_t \end{bmatrix} + b_i$ with $\hat{\theta}_i'$ and b_i denoting the weights and bias of the network respectively. Given that the current active mode is i with probability $\mu_t^i = 1$, the i-th network's weights and biases correspond to the parameters θ_i of the affine submodel in (1). The decoder output is the weighted sum of the individual network's outputs given by

$$\hat{y}_t = \sum_{i=1}^K \mu_t^i \theta_i' \begin{bmatrix} 1 \\ x_t \end{bmatrix}, \tag{7}$$

This is equivalent to the probability weighted affine (PrA) model (2) with probabilities given by the encoder. As noted before, such PrA model provides a smooth relaxation of PWA model in (1), such that the deterministic partition is replaced by probabilistic boundaries. This particular structure of the decoder along with categorical parameterization of the encoder distribution allows us to compute the expected likelihood over latent space analytically. Alternatively, in order to learn PWN and NPWN models, we can consider a *non-linear* decoder consisting of K independent feed-forward MLP neural networks having weights θ_i, with inputs x_t, μ_t^i and output $\hat{y}_t^i = \mathcal{NN}_d(x_t, \mu_t^i; \theta_i)$. Each network corresponds to local *non-linear* submodel and the decoder output is given by the probability weighted sum of K non-linear network outputs, $\hat{y}_t = \sum_{i=1}^K \mu_t^i \mathcal{NN}_d(x_t, \mu_t^i; \theta_i)$.

5. EVIDENCE LOWER BOUND LOSS

In order to train the VAE for piecewise models described above, in this section, we derive the ELBO loss $\mathcal{L}(\phi, \Theta)$ to be optimized over the encoder and decoder parameters.

Prior: Let us define the prior over the latent modes z_t as the following categorical distribution,

$$p(z_t|x_t) = \prod_{i=1}^K (\pi_t^i)^{z_t^i}, \tag{8}$$

where $\pi_t^i \in [0, 1]$ is the prior probability of the i-th mode, *i.e.*, $\pi_t^i = p(z_t^i = 1)$. Recall that z_t is K dimensional 1-of-K encoded categorical variable with the property that exactly one element has value 1 and the others have the value 0, thus, probability mass function $p(z_t|x_t) = p(z_t^i = 1)$ for the *active* mode i. This allows incorporating prior knowledge about the initial clustering pattern of the system. Possible choices for prior distribution are provided in the following.

Uniform prior: We can set $\pi_t^i = \frac{1}{K}$, *i.e.*, $p(z_t|x_t) = \frac{1}{K}$, to weigh all modes equally.

Dirichlet hyperprior: Let $\alpha = [\alpha_1 \ldots, \alpha_K]$ be concentration hyperparameter of a Dirichlet distribution $\mathrm{Dir}(K|\alpha)$. The prior probabilities are then given by

$$\pi_t = [\pi_t^1, \ldots, \pi_t^K] \sim \mathrm{Dir}(K|\alpha) \sim \frac{1}{B(\alpha)} \prod_{i=1}^{K} (\pi_t^i)^{\alpha_i - 1},$$

The choice of hyperparameter α determines the priorities assigned to a specific mode.

Likelihood: For PWA model structure, the *likelihood* of the output observation at time t is given as follows

$$\begin{aligned} p_\Theta(y_t|z_t, x_t) = \; & p_{\theta_i}(y_t|x_t, z_t^i = 1) = \mathcal{N}(y_t; \theta_i^{'} \left[\begin{smallmatrix} 1 \\ x_t \end{smallmatrix}\right], \sigma_v^2 I_{n_y}) \\ & = \prod_{i=1}^{K} (\mathcal{N}(y_t; \theta_i^{'} \left[\begin{smallmatrix} 1 \\ x_t \end{smallmatrix}\right], \sigma_v^2 I_{n_y}))^{z_t^i}, \end{aligned} \tag{9}$$

where we recall, σ_v^2 is the variance of the measurement noise. Since the noise samples v_t in (4) are assumed to be i.i.d., the overall likelihood is composed of product over the likelihood of individual observations given as follows,

$$\begin{aligned} & p_\Theta(y_1, \cdots, y_T|z_1, \cdots, z_T, x_1, \cdots, x_N) \\ & = \prod_{t=1}^{T} \prod_{i=1}^{K} (\mathcal{N}(y_t; \theta_i^{'} \left[\begin{smallmatrix} 1 \\ x_t \end{smallmatrix}\right], \sigma_v^2 I_{n_y}))^{z_t^i} \end{aligned} \tag{10}$$

The likelihood for PWN and NPWN model class (3) can be derived in a similar manner.

Variational distribution: The encoder provides a distribution over discrete latent variable z_t parameterized as the following categorical distribution,

$$q_\phi(z_t|x_t) = \prod_{i=1}^{K} (\mu_t^i)^{z_t^i} \tag{11}$$

where $\mu_t^i \in [0, 1], \quad i = 1, \ldots K$ are the probabilities given by the encoder output (see (6)) and ϕ are the weights of the encoder network. Note that the true (unknown) posterior $p_\Theta(z_t|x_t, y_t)$ is approximated by the variational distribution $q_\phi(z_t|x_t)$ given by the encoder.

Evidence lower bound: Next, we derive the expression for the evidence lower bound. We recall that the Kullback–Leibler (KL) divergence between the distribution $q_\phi(z_t|x_t)$, given by the probabilistic encoder network and the true (unknown) posterior $p_\Theta(z_t|y_t, x_t)$ can be written as,

$$\begin{aligned} & \mathrm{D_{KL}}(q_\phi(z_t|x_t)||p_\Theta(z_t|x_t, y_t)) \\ & = \log(p_\Theta(y_t|x_t)) + \mathrm{D_{KL}}(q_\phi(z_t|x_t)||p(z_t|x_t)) \\ & \quad - \mathbb{E}_{z_t \sim q_\phi(z_t|x_t)} [\log p_\Theta(y_t|z_t, x_t)] \end{aligned} \tag{12}$$

where $p_\Theta(y_t|x_t)$ is the *marginal* likelihood or the *evidence*. Note that the left hand side of (12) is the KL divergence between the two distributions, it is always non-negative. Thus, the evidence is lower bounded by the following term,

$$\begin{aligned} \log(p_\Theta(y_t|x_t)) \geq \; & \mathbb{E}_{z_t \sim q_\phi(z_t|x_t)} [\log p_\Theta(y_t|z_t, x_t)] \\ & - \mathrm{D_{KL}}(q_\phi(z_t|x_t)||p(z_t|x_t)) \end{aligned} \tag{13}$$

The right hand side of (13) is the *evidence lower bound* (ELBO) of the data sample at t. Re-writing (12), we define the ELBO loss $\mathcal{L}_{\Theta,\phi}(x_t, y_t)$ as follows,

$$\mathcal{L}_{\Theta,\phi}(x_t, y_t)$$

$$= \log(p_\Theta(y_t|x_t)) - \mathrm{D_{KL}}(q_\phi(z_t|x_t)||p_\Theta(z_t|x_t, y_t)) \tag{14a}$$

$$= \mathbb{E}_{z_t \sim q_\phi(z_t|x_t)} [\log p_\Theta(y_t|z_t, x_t)] - \mathrm{D_{KL}}(q_\phi(z_t|x_t)||p(z_t|x_t)) \tag{14b}$$

Note that from (14a), maximizing the ELBO loss $\mathcal{L}_{\Theta,\phi}$ implies maximization of the log marginal likelihood $\log(p_\Theta(y_t|x_t))$ for model learning from data and forcing the approximate posterior $q_\phi(.)$ towards the true one by minimizing the KL distance $\mathrm{D_{KL}}(q_\phi(\cdot)||p_\Theta(z_t|x_t, y_t)$, in order to have regular latent structure. However, both of these terms are intractable to compute. Nonetheless, this objective is achieved by considering the ELBO loss given in (14b), where the constitutive terms can be computed analytically. Thus, the goal is to maximize the ELBO loss $\mathcal{L}_{\Theta,\phi}$ given in (14b) w.r.t. both the encoder parameters ϕ and the decoder parameters Θ.

We now compute both the terms of the loss $\mathcal{L}_{\Theta,\phi}$ in (14b) for PWA model.

Reconstruction error: The first term in (14b) is the averaged log likelihood of the model over the approximate posterior distribution. This term is equivalent to "reconstruction or fitting error" in an autoencoder which drives learning of the model from data.

Substituting the expression of the likelihood (9), we have

$$\begin{aligned} & \mathbb{E}_{z_t \sim q_\phi(z_t|x_t)} [\log p_\Theta(y_t|z_t, x_t)] \\ & = \mathbb{E}_{z_t \sim q_\phi(z_t|x_t)} \left[\log \prod_{i=1}^{K} (\mathcal{N}(y_t; \theta_i^{'} \left[\begin{smallmatrix} 1 \\ x_t \end{smallmatrix}\right], \sigma_v^2 I_{n_y}))^{z_t^i} \right] \\ & = \mathbb{E}_{z_t \sim q_\phi(z_t|x_t)} \left[\sum_{i=1}^{K} z_t^i \log(\mathcal{N}(y_t; \theta_i^{'} \left[\begin{smallmatrix} 1 \\ x_t \end{smallmatrix}\right], \sigma_v^2 I_{n_y})) \right] \\ & = \mathbb{E}_{z_t \sim q_\phi(z_t|x_t)} \left[\frac{-1}{2\sigma_v^2} \sum_{i=1}^{K} z_t^i \left\| y_t - \theta_i^{'} \left[\begin{smallmatrix} 1 \\ x_t \end{smallmatrix}\right] \right\|^2 + \mathrm{const} \right] \\ & = \frac{-1}{2\sigma_v^2} \sum_{i=1}^{K} \mu_t^i \left\| y_t - \theta_i^{'} \left[\begin{smallmatrix} 1 \\ x_t \end{smallmatrix}\right] \right\|^2 + \mathrm{const} \end{aligned} \tag{15}$$

Given the categorical approximate posterior distribution $q_\phi(z_t|x_t)$ of the encoder in (11), the last equality in (15) is obtained using the fact that $\mathbb{E}_{z_t \sim q(z_t|x_t)} \left[z_t^i\right] = \mu_t^i$ for the categorical variable z_t.

Equivalently, for the probability weighted affine model (PrA) of the decoder, we have,

$$\mathbb{E}_{z_t \sim q_\phi(z_t|x_t)} [\log p_\Theta(y_t|z_t, x_t)] = \frac{-1}{2\sigma_v^2} \left\| y_t - \sum_{i=1}^{K} \mu_t^i \theta_i^{'} \left[\begin{smallmatrix} 1 \\ x_t \end{smallmatrix}\right] \right\|^2, \tag{16}$$

and for piecewise non-linear models we have,

$$\begin{aligned} & \mathbb{E}_{z_t \sim q_\phi(z_t|x_t)} [\log p_\Theta(y_t|z_t, x_t)] = \\ & \frac{-1}{2\sigma_v^2} \left\| y_t - \sum_{i=1}^{K} \mu_t^i \mathcal{NN}_d(x_t, \mu_t^i; \theta_i) \right\|^2 \end{aligned} \tag{17}$$

where $\mathcal{NN}_d(\cdot)$ is the non-linear MLP decoder network.

Regularization over latent variables: The second term in (14b) is KL divergence between the (approximate) posterior (11) and prior over latent variables defined in (8). This acts as a regularization term and provides a consistent structure to the latent space. Substituting (8) and (11), in the second term of (14b) we have,

$$\mathrm{D_{KL}}(q_\phi(z_t|x_t)||p(z_t|x_t)) = \mathbb{E}_{z_t\sim q_\phi(z_t|x_t)}\left[\log\left(\frac{q_\phi(z_t|x_t)}{p(z_t|x_t)}\right)\right]$$
$$= \mathbb{E}_{z_t\sim q_\phi(z_t|x_t)}\left[\log\left(\frac{\prod_{i=1}^K(\mu_t^i)^{z_t^i}}{\prod_{i=1}^K(\pi_t^i)^{z_t^i}}\right)\right]$$
$$= \mathbb{E}_{z_t\sim q_\phi(z_t|x_t)}\left[\sum_{i=1}^K z_t^i\log(\mu_t^i) - z_t^i\log(\pi_t^i)\right]$$
$$= \sum_{i=1}^K \mu_t^i\log\left(\frac{\mu_t^i}{\pi_t^i}\right) \tag{18}$$

The last equality follows as $\mathbb{E}_{z_t\sim q_\phi(z_t|x_t)}\left[z_t^i\right] = \mu_t^i$.

For a uniform prior, we have $\pi_t^i = \frac{1}{K}$, thus the KL term simplifies to

$$\sum_{i=1}^K \mu_t^i\log\left(\frac{\mu_t^i}{\pi_t^i}\right) = \sum_{i=1}^K \mu_t^i\log\left(\mu_t^i\right) - \log(1/K)$$

The first term can be interpreted as the (negative) *mode entropy* which allows to control potential mode collapse during the training. We note that $\lim_{\mu_t^i\to 0}\ \mu_t^i\log\left(\mu_t^i\right) = 0$, thus, the regularization term is set to 0 for non-active modes having small probability values.

Substituting the expressions of fitting error (15) and regularization loss (18) in the ELBO (14b) and averaging over all data samples, we get the following loss function

$$\mathcal{L}_{\Theta,\phi}(\{x_t, y_t\}_{t=1}^T) =$$
$$\frac{1}{T}\sum_{t=1}^T\sum_{i=1}^K\left(-\mu_t^i\left\|y_t - \theta_i'\begin{bmatrix}1\\ x_t\end{bmatrix}\right\|^2 - \lambda\mu_t^i\log\left(\frac{\mu_t^i}{\pi_t^i}\right)\right) \tag{19}$$

where λ is a regularization hyper-parameter to achieve trade-off between reconstruction error and regularization loss. The encoder and decoder networks are trained at once to compute the parameters ϕ, Θ via back-propagation of the loss (19) over training data. The estimate ϕ defines an encoder which approximates the posterior $p_\Theta(z_t|y_t, x_t)$ characterizing the partition of the regressor space, while decoder parameters Θ give the parameters of the local submodels of the piecewise hybrid model, thus, solving Problem 1.

6. NUMERICAL EXAMPLE

The effectiveness of the proposed technique is evaluated on numerical example, namely, identification of a NPWA-ARX system is presented. Further examples, e.g., PWA function regression, identification of a benchmark PWA-ARX system and identification of PWNL model using non-linear decoder networks, with link to the codes are reported in the technical report (Mejari et al., 2022). All computations are carried out on an i7 1.9-GHz Intel core processor with 32 GB of RAM. The codes are implemented with `PyTorch 1.12.1` for the training of the neural networks.

The quality of the trained models is assessed in terms of their ability to recognize the partition of the regressor space and their predictive capability quantified via R^2 score computed on a test dataset $R^2 = \left(1 - \frac{\sum_{t=1}^T(y_t-\hat{y}_t)^2}{\sum_{t=1}^T(y_t-\bar{y})^2}\right)\times 100$ %, where y is the measured output, $\hat{y}$ is the estimated model output and $\bar{y}$ is the average value of y, *i.e.*, $\bar{y} = \frac{1}{T}\sum_{t=1}^T y_t$. We have chosen a uniform prior $p(z_t) = \frac{1}{K}$. The number of hidden layers and number of nodes in each layer of encoder-decoder networks are tuning hyper-parameters of the VAE, which are chosen via cross-validation.

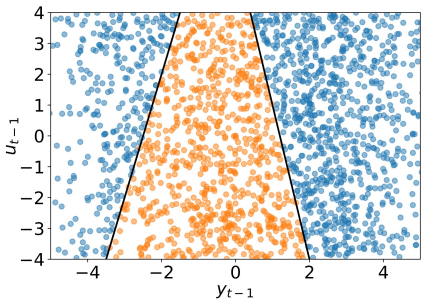

Fig. 2. NPWA-ARX model: True partition (solid black lines) *vs* estimated clustering of the regressor space.

6.1 Identification of NPWA-ARX model

In this example, we consider estimation of *nonlinearly* piecewise affine system. Specifically, we consider the following data-generating system as a slight modification to the benchmark PWA-ARX system (Bemporad et al., 2005), which can be conceived as an extension to the nonlinearly piecewise affine model (Lauer and Bloch, 2008; Brusaferri et al., 2020):

$$y_t = \begin{cases} \begin{bmatrix}-0.4 & 1 & 1.5\end{bmatrix}x_t + e_t, & \text{if } \begin{bmatrix}4 & -1 & 10\end{bmatrix}x_t < 0, \\ \begin{bmatrix}0.5 & -1 & -0.5\end{bmatrix}x_t + e_t, & \text{if } \begin{bmatrix}4 & -1 & 10\end{bmatrix}x_t \geq 0, \\ & \quad \& \begin{bmatrix}5 & 1 & -6\end{bmatrix}x_t \leq 0, \\ \begin{bmatrix}-0.4 & 1 & 1.5\end{bmatrix}x_t + e_t, & \text{if } \begin{bmatrix}5 & 1 & -6\end{bmatrix}x_t > 0, \end{cases} \tag{20}$$

with regressor $x_t = \begin{bmatrix}y_{t-1} & u_{t-1} & 1\end{bmatrix}'$ and $e \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 0.3$, which corresponds to the SNR = 19 dB. A training dataset of 6000 samples and a noise-free test dataset of 2000 samples is gathered. Here, the system with $K = 2$ local affine models is assumed to introduce *nonlinear* partitioning requirements. We remark that the problem could be solved by considering a PWA-ARX model with $K = 3$ modes and *linearly separable* clusters. However, in this example, we aim to infer nonlinearly PWA-ARX (NPWA-ARX) model by considering only $K = 2$ modes with a nonlinear boundary between mode 1 and mode 2. Note that the system is defined by the same dynamics occurring over two regions of the regressor space, although, identification methods which rely on *linear* partition of the regression space would require three modes as the regions are not linearly separable. For estimation with VAE, we consider encoder network having a single-hidden-layer with *relu* activations in the hidden layer and *softmax* activation at the output layer. The number of nodes in the hidden layer of the encoder is set to 10, while the number of output layer nodes is set to $K = 2$. The decoder consists of 2 *linear* networks, each with a single layer having dimension equal to the dimension of the parameters of local affine function in (20). The VAE is trained by maximizing the loss function (19) with $\lambda = 1\cdot 10^{-3}$. The learning rate is set to $1\cdot 10^{-3}$ and number of SGD iterations are fixed to $20\cdot 10^3$. The

Table 1. True *vs* estimated model parameters (weights and biases of the decoder network).

Mode	True	Estimated
1	$[-0.4,\ 1,\ 1.5]$	$[-0.4032,\ 1.0006,\ 1.5034]$
2	$[0.5,\ -1,\ -0.5]$	$[0.5047,\ -1.0012,\ -0.5054]$

required training time is 92.5 sec. The estimated clustering pattern is shown in Fig. 2. It can be seen from the figure that despite nonlinear partitioning induced by the data-generating system, the encoder is able to recognize the underlying operating regions of the two dynamics very accurately. The parameters of the local affine models (decoder weights and biases) are reported in Table 1, which closely match the true system parameters for both modes. Finally, the R^2 score obtained on the test data computed using *one-step-ahead* predicted output is $R^2\ (1-\text{step}) = 99.68\%$ and using *simulated* output $R^2\ (\text{sim}) = 96.77\%$, which shows the estimated model is able to reconstruct the output with high accuracy.

7. CONCLUSION AND FUTURE WORKS

We have presented a framework for learning piecewise models using specialized variational autoencoder. In contrast to the traditional black-box deep learning models, the developed VAE is interpretable, in the sense that the latent space can be interpreted in terms of the modes of the underlying hybrid system while the decoder represents local submodels. The developed approach is effective to identify a general class of piecewise models as demonstrated in numerical case study. Future works involve extension of the proposed method to PWA state-space models and investigating other variants of VAE, *e.g.*, vector quantized (VQ-VAE) for data-driven modeling of hybrid systems.

REFERENCES

Bemporad, A., Garulli, A., Paoletti, S., and Vicino, A. (2005). A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50(10), 1567–1580.

Bemporad, A. (2022). A piecewise linear regression and classification algorithm with application to learning and model predictive control of hybrid systems. *IEEE Transactions on Automatic Control*, 1–16. doi:10.1109/TAC.2022.3183036.

Boukharouba, K., Bako, L., and Lecoeuche, S. (2009). Identification of piecewise affine systems based on dempster-shafer theory. In *Proc. 15th IFAC Symposium on System Identification*, 1662–1667. Saint-Malo, France.

Breschi, V., Piga, D., and Bemporad, A. (2016). Piecewise affine regression via recursive multiple least squares and multicategory discrimination. *Automatica*, 73, 155–162.

Brusaferri, A., Matteucci, M., and Spinelli, S. (2020). Identification of probability weighted ARX models with arbitrary domains. *arXiv:2009.13975*.

Forgione, M. and Piga, D. (2020). Model structures and fitting criteria for system identification with neural networks. In *Proc. of the 14th IEEE International Conference on Application of Information and Communication Technologies (AICT)*, 1–6. Tashkent, Uzbekistan.

Garulli, A., Paoletti, S., and Vicino, A. (2012). A survey on switched and piecewise affine system identification. In *Proc. of the 16th IFAC Symposium on System Identification*, 344–355. Brussels, Belgium.

Juloski, A.L., Weiland, S., and Heemels, W.P.M.H. (2005). A Bayesian approach to identification of hybrid systems. *IEEE Transactions on Automatic Control*, 50(10), 1520–1533.

Kingma, D.P. and Welling, M. (2014). Auto-encoding variational Bayes. *arXiv:1312.6114v10*.

Lauer, F. (2015). On the complexity of piecewise affine system identification. *Automatica*, 62, 148–153.

Lauer, F. and Bloch, G. (2014). Piecewise smooth system identification in reproducing kernel hilbert space. In *53rd IEEE Conference on Decision and Control*, 6498–6503.

Lauer, F. and Bloch, G. (2008). Switched and piecewise nonlinear hybrid system identification. In *Hybrid Systems: Computation and Control*, 330–343. Springer, Berlin, Heidelberg.

Ljung, L., Andersson, C., Tiels, K., and Schön, T.B. (2020). Deep learning and system identification. In *Proc. of the 21st IFAC World Congress*, 1175–1181. Berlin, Germany.

Masti, D. and Bemporad, A. (2021). Learning nonlinear state–space models using autoencoders. *Automatica*, 129, 109666.

Mazzoleni, M., Breschi, V., and Formentin, S. (2021). Piecewise nonlinear regression with data augmentation. In *Proc. of the 19th IFAC Symposium on System Identification SYSID*, 421–426. Padova, Italy.

Mejari, M., Breschi, V., and Piga, D. (2020a). Recursive bias-correction method for identification of piecewise affine output-error models. *IEEE Control Systems Letters*, 4(4), 970–975.

Mejari, M., Forgione, M., and Piga, D. (2022). Variational autoencoder for the identification of piecewise models. *Technical Report*. URL `https://manasdm.github.io/publication/preprint_vae22/TR_IDSIA_vae_hybrid_id.pdf`.

Mejari, M., Naik, V.V., Piga, D., and Bemporad, A. (2020b). Identification of hybrid and linear parameter-varying models via piecewise affine regression using mixed integer programming. *International Journal of Robust and Nonlinear Control*, 30(15), 5802–5819.

Piga, D., Bemporad, A., and Benavoli, A. (2020). Rao-Blackwellized sampling for batch and recursive Bayesian inference of Piecewise Affine models. *Automatica*, 117, 109002.

Piga, D., Forgione, M., and Mejari, M. (2021). Deep learning with transfer functions: new applications in system identification. In *Proc. of the 19th IFAC Symposium System Identification SYSID*, 415–420. Padova, Italy.

Roll, J., Bemporad, A., and Ljung, L. (2004). Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1), 37–50.

Taguchi, S., Suzuki, T., Hayakawa, S., and Inagaki, S. (2009). Identification of probability weighted multiple ARX models and its application to behavior analysis. In *Proc. of the 48h IEEE Conf. on Decision and Control (CDC)*, 3952–3957. Shanghai, China.

Wang, Y. (2017). A new concept using LSTM neural networks for dynamic system identification. In *Proc. of the 2017 American Control Conference (ACC)*, 5324–5329. Seattle, WA, USA.