Dynamical Variational Autoencoders : discrete-time and continuous-time models. Links to stochastic calculus and stochastic differential equations

ENS Paris-Saclay, MVA

Benjamin Deporte : benjamin.deporte@ens-paris-saclay.fr
benjamin.deporte@polytechnique.org

September 2025

1 Remerciements

Avec ce travail sur les VAEs dynamiques se tourne une page significative de mon parcours en machine learning.

Un grand merci à Pierre-Alexandre Mattei de l'INRIA, qui a accepté d'être mon enseignant référent pour cette aventure ! Toujours disponible, bienveillant et patient devant les pires de mes questions, avec des idées et l'expérience pour me débloquer aux moments clefs ! Merci également à Pierre Latouche, avec lequel Pierre-Alexandre a assuré le cours "Introduction to Probabilistic Graphical Models and Deep Generative Models" au 1er trimestre du MVA.

Merci également à l'équipe du MVA : direction, enseignants, administration. Le MVA est un Master de qualité, j'y ai appris énormément de nouvelles choses et consolidé des connaissances plus anciennes. L'enseignement y est bienveillant sans être laxiste, exigeant et productif. Une belle formation, et à titre personnel une belle aventure Francilienne pendant quelques mois ! Note à Laurent Oudre : je parle un peu de séries temporelles dans ce rapport, pas mal de calcul stochastique, mais pas un mot sur les maths financières, promis !

A mes collègues de l'IRT : merci Lionel ! Me permettre de cumuler mon poste et le MVA pendant une petite année a représenté beaucoup pour moi. Toutes les organisations et tous les managers ne l'auraient pas permis, loin de là. Aux collègues du 6e étage du B612 - Greg, Franck, Lucas, Sebastien... : oui ma densité de probabilité a chuté, mais elle s'apprête à remonter ! Ne revendez pas tout de suite le mobilier de bureau. A bientôt.

A mon épouse Anne-Laure, si jamais elle lit ce rapport pour s'endormir : j'étais sincère -enfin, en moyenne- quand je disais tenter le MVA en deux ans parce que ce serait le plus raisonnable en terme d'emploi du temps. Mais la logique Bayesienne, après tout, est de changer d'avis en fonction des données disponibles. Et avouons-le, je n'ai jamais été du genre patient. Merci à toi.

A Jean-Michel Loubes - qui m'a mis le pied à l'étrier il y a quelques temps déjà, pointé vers les bons textes de référence, fait rencontrer beaucoup de monde pour valider ce qui, à l'époque, n'était qu'une piste d'évolution professionnelle parmis d'autres. Revenir aux maths, et venir au machine learning, aura été un de mes meilleurs choix professionnels et personnels. J'ai retrouvé une envie intellectuelle que je n'avais plus connue depuis très longtemps : merci à toi. Merci également de m'avoir accepté en MAPI3 à Paul Sabatier il y a de cela quelque temps maintenant.

J'oublie probablement du monde, et je m'en excuse par avance. Merci en tout cas de m'avoir accompagné.

Place aux VAEs dynamiques.

Contents

List of Figures

Part I

Introduction

Variational Auto Encoders (VAEs) are a well-known class of generative models, described in [11], which have spawn numerous applications. However, VAEs posit an i.i.d. assumption over the latent variables, that carries strong limitations if considering data sequences over time, where correlations often exist between data samples. A richer class of models aims at solving this limitation : the Dynamical Variational Auto Encoders (DVAEs). In DVAEs, the latent variables are structured themselves as a correlated set (usually a sequence also), aiming at encoding the temporal behavior of the data. The relationship between the latent variables and the observed data (ie, the observation model, or decoder) remains as in regular VAEs.

The first part of the report is dedicated to the general study of DVAEs. We rely on the exhaustive survey [10], which is the basis for this part. We review the general formulation of DVAEs, and review the detailed implementation of two discrete time models: the Deep Kalman Filter (DKF), and the Variational Recurrent Neural Network (VRNN). A discrete-time setting of the latent variables is straightforward : for example, structuring the latent variables as a first-order Markov chain leads to the general framework of State Space Models (SSMs). However, this setting carries limitations too. The first limitation, and possibly most important one, is that the structure of the model requires regularly-sampled data. In practice, data can be observed with a changing frequency. A natural idea is then to structure the latent variables as a continuous-time process, which will be sampled if and when required to match an irregularly sampled data. A candidate for such a continuous-time process, endowed with convenient properties, is the Gaussian Process (GP) [22]. We review the corresponding DVAE : the Gaussian Process Variational Auto Encoder (GP-VAE) model. This model is only briefly mentionned in [10] (where the focus is on discrete time models), but described in more details in [2], [8], [27] and [28].

The second part describes the implementations of those three models, the experiments run on toy datasets, the PyTorch tricks that sometimes cost quite a bit to learn (...), and the take-aways. A repository with the set of code and notebooks can be found at Benjamin's GitHub repo.

One of the GP-VAE papers, specifically [28], has triggered a significant interest in me, as it showed a deep relationship between DVAEs and stochastic calculus.

I then invested a significant amount of time in studying the mathematical machinery required to fully understand this relationship. This steered me a little bit off course the subject per say, but the theory of stochastic calculus proved fascinating and commendable. Among good study books, are [16] and [25]. Getting up to speed on stochastic calculus has been quite a challenge given the time frame but proved extremely useful. We present some key results, which the knowledgeable reader on stochastic calculus can skip, and added a longer reference at the end of this report. The aficionado of generative models will recognize the mathematical framework of diffusion models.

A first relationship between DVAEs and stochastic calculus is as follows: the solution of a general Stochastic Differential Equation (SDE) is a Markov process (where the evolution over time of the transition probability density is described by the Fokker-Plank-Kolmogorov equation). This Markov process, when discretized, leads to the natural framework of our first two discrete models : the DKF and VRNN. Moreover, in the case of a linear SDE, the solution of the SDE is also a Gaussian process, which points to GP-VAE. For solutions of linear SDE, one can use off the shelf smoothers (and filters), such as Kalman and Rauch-Tung-Striebel (RTS), that scale linearly instead of cubically as is usually the case for GPs ([28], [25])

However, the GP-VAE model carries limitations too. First, some Gaussian processes -more specifically some kernel functions- can not be formulated as the solution to a linear SDE. We review quickly some such kernels. Second, some stochastic processes are both Markov processes and Gaussian processes (the Brownian motion itself, but also the well-studied Ornstein Uhlenbeck process), but some Markov processes (ie solutions to general SDE) are not Gaussian processes.

An idea is then to use a more general stochastic process, defined as the solution to a general (ie non-linear) SDE, as the model for latent variables. The drift and the diffusion can also be learnt as neural networks. This is the field of Neural Stochastic Differential Equations (Neural-SDEs) ([19]), which is a recent research field, and builds upon the Neural Ordinary Differential Equations (Neural-ODEs) ([3], [24]) framework. We present some of the ideas of

Neural-ODEs: viewing Neural-ODE models as continuous time ResNet models; using the adjoint method and a backward Ordinary Differential Equation (ODE) to compute gradients ([14]).

We have lacked time to go much further... for now.

It's been quite a journey...

Part II

Dynamical Variational Autoencoders - Theory

This part presents the general framework of DVAEs, based on [10].

We start with a reminder of the key notion of **D-separation**, which is central in deriving conditional probabilities of DVAEs from their associated Graphical Probabilistic Models (GPMs).

Then, we describe three models in details:

- **Deep Kalman Filter** : this model arises as the first evolution of the well-known Kalman Filter, with richer, Multi Layer Perceptrons (MLPs) networks for the encoder and decoder.
- **Variational Recurrent Neural Network** : at the other end of the spectrum, VRNNs provide the most expressive discrete-time formulation of the encoder and decoder. We describe here a different implementation from the one described in [10].
- **Gaussian Process Variational Auto Encoder** : in this model, the prior over the latent variables is no longer discrete, but is a GP. This allows for sampling data at irregular intervals. Another benefit is the use of richer kernel families to encode prior knowledge.

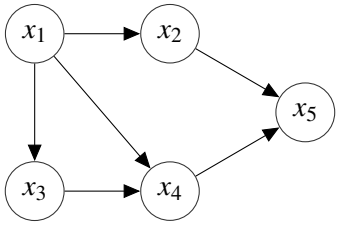
2 D-separation

Graphical models are an efficient way to describe families of factorized joint distributions of a data set $(x_i)_{i=1,...,n}$ into a Directed Acyclic Graph (DAG).

Given such a dataset, we can build a DAG where each node is indexed by an integer that is higher than the indexes of its *parent* nodes, such that the joint distribution over the dataset factorizes as:

$$p(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} p(x_i|pa_i) \tag{2.1}$$

where pa_i is the set of parent nodes of x_i.
For example, the following DAG



describes:

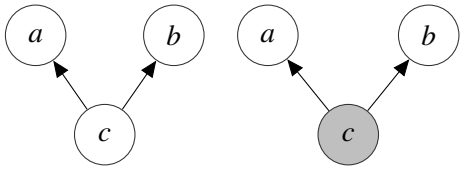
$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_1, x_3)p(x_5|x_2, x_4) \tag{2.2}$$

Describing a factorized joint probability distribution by a DAG allows to determine graphically whether two sets of nodes (ie random variables) are independent, conditioned on a third set of nodes. This allows subsequently to simplify the expressions of the observation model $p(x|z)$, and/or of the posterior model $q(z|x)$.

D-Separation is the set of rules that determine whether there is conditional independence between two sets given a third one. D-Separation is well described in key books such as [1], [13] or [17]. We will enunciate here the key concepts, and refer the interested reader to those books.

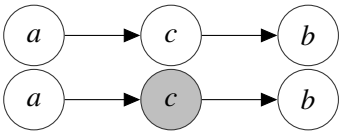
D-Separation is a way to find out graphically conditional (in)dependence relationships between random variables, that would be more difficult to calculate by marginalizing the joint distribution over the conditioning variables. A nice way to demonstrate this, is to review the three examples of 3-node DAG. NB : The observed (ie conditioning) variables are noted with gray background.

Example 1 : c is said *tail-to-tail* with a and b, and *blocks the path between a and b*, making them conditionally independent : $a \not\perp\!\!\!\perp b \,|\, \varnothing$, $a \perp\!\!\!\perp b \,|\, c$



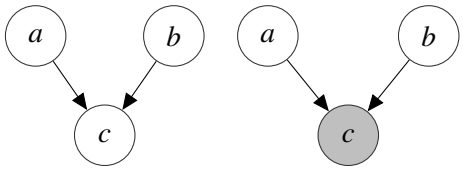
$$
\begin{aligned}
p(a,b,c) &= p(c)p(a|c)p(b|c) \\
p(a,b) &= \sum_c p(c)p(a|c)p(b|c) \neq p(a)p(b) \\
&\implies a \not\perp\!\!\!\perp b \,|\, \emptyset \\
p(a,b|c) &= p(a|c)p(b|c) \\
&\implies a \perp\!\!\!\perp b \,|\, c
\end{aligned}
$$

Example 2 : a, c, b form a Markov chain. c is said *head-to-tail* with a and b and, here also, *blocks the path between a and b*, making them conditionally independent : $a \not\perp\!\!\!\perp b \,|\, \emptyset$, $a \perp\!\!\!\perp b|c$



$$
\begin{aligned}
p(a,b,c) &= p(a)p(c|a)p(b|c) \\
p(a,b) &\neq p(a)p(b) \implies a \not\perp\!\!\!\perp b|\emptyset \\
p(a,b|c) &= \frac{p(a)p(c|a)}{p(c)}p(b|c) = p(a|c)p(b|c) \implies a \perp\!\!\!\perp b \,|\, c
\end{aligned}
$$

Example 3 : c is said *head-to-head* with a and b. In this head-to-head configuration, contrary to the two examples above, *the path between a and b is blocked when c is unobserved* : $a \perp\!\!\!\perp b \,|\, \emptyset$, $a \not\perp\!\!\!\perp b \,|\, c$



$$
\begin{aligned}
p(a,b,c) &= p(a)p(b)p(c|a,b) \\
p(a,b) &= \sum_c p(a)p(b)p(c|a,b) = p(a)p(b) \\
&\implies a \perp\!\!\!\perp b \,|\, \emptyset \\
p(a,b|c) &\neq p(a|c)p(b|c) \\
&\implies a \not\perp\!\!\!\perp b \,|\, c
\end{aligned}
$$

We can extend the notion to full sets of nodes.

D-Separation

Let $\mathcal{G}$ be a DAG.
Let A, B, C three disjoint sets of nodes in $\mathcal{G}$: $A \cap B = A \cap C = B \cap C = \emptyset$.
C is the set of "conditioning nodes", or "observed variables".
We aim to determine whether $A \perp\!\!\!\perp B \mid C$.
Algorithm

1. **Evaluate each path between A and B**
   Evaluate each possible path between any point $a \in A$, and any point $b \in B$. Such a path between a and b is said **blocked** if it contains one node n such that one of two following conditions is true:
   - arrows in the path are *head-to-tail* or *tail-to-tail* at node n, and $n \in C$ (n is an observed/conditioning node).
   - arrows in the path are *head-to-head* at node n, and $n \notin C$ and none of n descendants is in C
2. **Assess all paths**
   - If all paths $(a, b), a \in A, b \in B$ are blocked, then A is said **D-separated** from B by C, and the joint distribution defined by $\mathcal{G}$ verifies $A \perp\!\!\!\perp B \mid C$.
   - If there is at least one path $(a, b), a \in A, b \in B$ that is not blocked then $A \not\perp\!\!\!\perp B \mid C$.

3 Dynamical Variational Auto Encoders

VAE models are well known and documented (see for example the seminal paper [11]. (A self-contained brief summary of VAE can be found in appendix A).

When dealing with sequential data, the i.i.d assumption on latent variables z_i is a limitation. By D-separation, all x_i's are independent of each other conditionally by z_i : $p(x_i|x_1, x_2, ..., x_{i-1}, x_{i+1}, .., x_n, z_i) = p(x_i|z_i)$. Therefore, a vanilla VAE can not account for correlations between x_i across time.

DVAEs encode a temporal dependency in the latent variables prior distribution. In this chapter, we review the general discrete-time setting, where the latent variables are countable and indexed by time. An exhaustive review of discrete-time DVAEs can be found in [10].

We start by some notations.

Notations

- the data is a sequence of T points noted $x_{1:T} = \{(x_t)_{t=1,...,T}\} \in \mathbb{R}^F$.
- the sequence of the associated T latent variables is $z_{1:T} = \{(z_t)_{t=1,...,T}\} \in \mathbb{R}^L$
- optionally, there may be a sequence of -usually deterministic- T inputs $u_{1:T} = \{(u_t)_{t=1,...,T}\} \in \mathbb{R}^U$

The generative model is given by the general expression of the joint distribution (here with a sequence of inputs) $p(x_{1:T}, z_{1:T}|u_{1:T})$:

$$\begin{aligned} p(x_{1:T}, z_{1:T}|u_{1:T}) &= \prod_{t=1}^{T} p(x_t, z_t|x_{1:t-1}, z_{1:t-1}, u_{1:T}) \\ &= \prod_{t=1}^{T} p(x_t|x_{1:t-1}, z_{1:t}, u_{1:T}) p(z_t|x_{1:t-1}, z_{1:t-1}, u_{1:T}) \\ &= \prod_{t=1}^{T} p(x_t|x_{1:t-1}, z_{1:t}, u_{1:t}) p(z_t|x_{1:t-1}, z_{1:t-1}, u_{1:t}) \end{aligned}$$

where the only assumption that is made is a causal dependency of the x_t, z_t on the inputs $u_{1:t}$, thus allowing to change the conditioning $|u_{1:T}$ into $|u_{1:t}$

In the rest of the report, we will consider systems with no input, and drop the conditioning on $u_{1:t}$ to simplify notations. However, the reasoning remains the same with inputs.

The true posterior $p(z_{1:T}|x_{1:T})$ is usually untractable, but can be developed:

$$p(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T} p(z_t|z_{1:t-1}, x_{1:T})$$

It can be noted that the true posterior exhibits a dependence of z_t on *past* $z_{1:t-1}$, but a dependence on the *whole* data sequence $x_{1:T}$ (think Kalman smoother).

As in vanilla VAEs, the inference model is the approximation of the true posterior by an parametric encoder $q_\phi(z_{1:T}|x_{1:T})$, where ϕ is the set of parameters:

$$q_\phi(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T} q_\phi(z_t|z_{1:t-1}, x_{1:T})$$

Depending on the chosen graphical models and the corresponding D-separation results, the observation model $p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}, u_{1:t})$ (with θ_x the set of parameters of the observation model) and approximate posterior $q_\phi(z_t|z_{1:t-1}, x_{1:T})$ may simplify.

It is also considered a good practice ([10]) to copy/paste the expression of $q_\phi(z_t|z_{1:t-1}, x_{1:T})$ from the expression of the true posterior resulting from the D-separation analysis (see next chapters for examples).

Equipped with the generative model and the inference model, we compute the log likelihood of the data $x_{1:T}$ and derive an Variational Lower Bound (VLB) for training (using the same manipulation as for vanilla VAE : multiplying both sides of the equation by q_ϕ and integrating over $dz_{1:T}$)

$$\log p(x_{1:T}) = \log \frac{p(x_{1:T}, z_{1:T})}{p(z_{1:T}|x_{1:T})} \tag{3.1}$$

$$= \mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})} \log \frac{p(x_{1:T}, z_{1:T})}{q_\phi(z_{1:T}|x_{1:T})} \frac{q_\phi(z_{1:T}|x_{1:T})}{p(z_{1:T}|x_{1:T})} \tag{3.2}$$

$$= \mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})} \log \frac{p(x_{1:T}, z_{1:T})}{q_\phi(z_{1:T}|x_{1:T})} + \mathbb{KL}\Big(q_\phi(z_{1:T}|x_{1:T})||p(z_{1:T}|x_{1:T})\Big) \tag{3.3}$$

$$\geq \mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})} \log \frac{p(x_{1:T}, z_{1:T})}{q_\phi(z_{1:T}|x_{1:T})} = \mathcal{L}(\theta, \phi, X) \tag{3.4}$$

The dependence of $\mathcal{L}(\theta, \phi, X)$ on θ is made more obvious when developing $\mathcal{L}(\theta, \phi, X)$.

Remember we have (making the set of parameters explicit) :

$$p_\theta(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t-1}) \tag{3.5}$$

$$q_\phi(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T} q_\phi(z_t|z_{1:t-1}, x_{1:T}) \tag{3.6}$$

Therefore

$$\mathcal{L}(\theta, \phi, X) = \mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})} \log \left(\frac{\prod_{t=1}^{T} p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t-1})}{\prod_{t=1}^{T} q_\phi(z_t|z_{1:t-1}, x_{1:T})} \right) \tag{3.7}$$

$$= \mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})} \left(\sum_{t=1}^{T} \log p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) - \sum_{t=1}^{T} \log \frac{q_\phi(z_t|z_{1:t-1}, x_{1:T})}{p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t-1})} \right) \tag{3.8}$$

At this point, the expectations require some work. First, we note that, as q_ϕ develops as 3.6, for any function Ψ, the first expectation can be written (note the change in indexes of z)

$$\mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})}\Psi(z_{1:t}) = \mathbb{E}_{q_\phi(z_{1:t}|x_{1:T})}\Psi(z_{1:t})$$

Second, we develop further and write:

$$\begin{aligned}\mathbb{E}_{q_\phi(z_{1:T}|x_{1:T})}\Psi(z_{1:t}) &= \mathbb{E}_{q_\phi(z_{1:t}|x_{1:T})}\Psi(z_{1:t}) \\ &= \mathbb{E}_{q_\phi(z_{1:t-1}|x_{1:T})}\mathbb{E}_{q_\phi(z_t|z_{1:t-1},x_{1:T})}\Psi(z_{1:t})\end{aligned}$$

Therefore the VLB becomes:

$$\mathcal{L}(\theta,\phi,X) = \mathbb{E}_{q_\phi(z_{1:t}|x_{1:T})}\sum_{t=1}^{T}\log p_{\theta_x}(x_t|x_{1:t-1},z_{1:t}) - \sum_{t=1}^{T}\mathbb{E}_{q_\phi(z_{1:t-1}|x_{1:T})}\left[\mathbb{E}_{q_\phi(z_t|z_{1:t-1},x_{1:T})}\log\frac{q_\phi(z_t|z_{1:t-1},x_{1:T})}{p_{\theta_z}(z_t|z_{1:t-1},x_{1:t-1})}\right] \tag{3.9}$$

$$= \sum_{t=1}^{T}\mathbb{E}_{q_\phi(z_{1:t}|x_{1:T})}\log p_{\theta_x}(x_t|x_{1:t-1},z_{1:t}) - \sum_{t=1}^{T}\mathbb{E}_{q_\phi(z_{1:t-1}|x_{1:T})}\mathbb{KL}\left(q_\phi(z_t|z_{1:t-1},x_{1:T})||p_{\theta_z}(z_t|z_{1:t-1},x_{1:t-1})\right) \tag{3.10}$$

As for the vanilla VAE, the VLB contains two terms.

- The first term is the reconstruction error. it is the sum over the time steps, of the average log likelihood the data at time t, given the approximate distribution of the past and present latent variables, and the past data.
- The second term is a regularization term, summing over the time steps the average divergence between the approximate posterior distribution of the latent variable at time t, and its real distribution.

As in vanilla VAE, the sampling over q_ϕ requires the use of the "re parametrization trick" (see [11]), for $\mathcal{L}(\theta,\phi,X)$ to be differentiable w.r.t. θ,ϕ.

Here is the summary regarding DVAE:

General Dynamical VAEs

- **generative model**

$$p(x_{1:T},z_{1:T}) = \prod_{t=1}^{T}p_{\theta_x}(x_t|x_{1:t-1},z_{1:t})p_{\theta_z}(z_t|z_{1:t-1},x_{1:t-1}) \tag{3.11}$$

- **inference model**

$$q_\phi(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T}q_\phi(z_t|z_{1:t-1},x_{1:T}) \tag{3.12}$$

- **VLB for training**

$$\begin{aligned}\mathcal{L}(\theta,\phi,X) = &\sum_{t=1}^{T}\mathbb{E}_{q_\phi(z_{1:t}|x_{1:T})}\log p_{\theta_x}(x_t|x_{1:t-1},z_{1:t}) \\ &- \sum_{t=1}^{T}\mathbb{E}_{q_\phi(z_{1:t-1}|x_{1:T})}\mathbb{KL}\left(q_\phi(z_t|z_{1:t-1},x_{1:T})||p_{\theta_z}(z_t|z_{1:t-1},x_{1:t-1})\right)\end{aligned} \tag{3.13}$$

4 Deep Kalman Filter

The Kalman Filter is a well known model, widely used to denoise time series observations and make predictions. The latent variables form a Markov Chain, and all the probability distributions (ie encoder, decoder and transition model) are linear Gaussians. This allows to derive close form expressions for the solutions (the well-known Kalman filter and Kalman smoother).

In a **Deep Kalman Filter**, the temporal structure of the latent variables is still a Markov Chain. The probaility models are still Gaussians, but with parameters mean and covariance learnt by neural networks.

More specifically, the DAG describing a Deep Kalman Filter is:

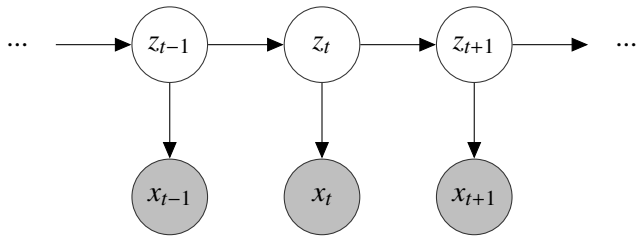


Figure 4.1: Probabilistic model of a Deep Kalman Filter

It is then particularly useful to use D-separation on the DAG to simplify the general DVAE expressions 3.11 and 3.12. Conditioning on z_t and z_{t-1} drives:

$$p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) = p_{\theta_x}(x_t|z_t) \tag{4.1}$$
$$p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t}) = p_{\theta_z}(z_t|z_{t-1}) \tag{4.2}$$
$$q_\phi(z_t|z_{1:t-1}, x_{1:T}) = q_\phi(z_t|z_{t-1}, x_{t:T}) \tag{4.3}$$

We then choose Gaussian distributions for p_{θ_x}, p_{θ_z} and q_ϕ, with mean and diagonal covariance, learnt by neural networks.

$$p_{\theta_x}(x_t|z_t) = \mathcal{N}(x_t|\mu_{\theta_x}(z_t), \operatorname{diag} \sigma^2_{\theta_x}(z_t)) \tag{4.4}$$
$$p_{\theta_z}(z_t|z_{t-1}) = \mathcal{N}(z_t|\mu_{\theta_z}(z_{t-1}), \operatorname{diag} \sigma^2_{\theta_z}(z_{t-1})) \tag{4.5}$$
$$q_\phi(z_t|z_{t-1}, x_{t:T}) = \mathcal{N}(z_t|\mu_\phi(z_{t-1}, x_{t:T}), \operatorname{diag} \sigma^2_{\theta_z}(z_{t-1}, x_{t:T})) \tag{4.6}$$

Some other formulations of the approximate posterior (encoder) are possible. For example:

$$q_\phi(z_t|z_{t-1}, x_t)$$
$$q_\phi(z_t|z_{1:t}, x_{1:t})$$
$$q_\phi(z_t|z_{1:T}, x_{1:T})$$

We have chosen 4.3 for the implementation, as it has the same formulation as the true posterior and respects the corresponding dependencies.

Taking note that:

$$q_\phi(z_{1:t}|x_{1:T}) = q_\phi(z_{1:t-1}|z_t, x_{1:T})q_\phi(z_t|x_{1:T})$$

And using D-Separation, the Evidence Lower Bound (ELBO) 3.13 simplifies into:

$$\mathcal{L}(\theta, \phi, X) = \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{1:t}|x_{1:T})} \log p_{\theta_x}(x_t|z_t) - \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{1:t-1}|x_{1:T})} \mathbb{KL}\left(q_\phi(z_t|z_{t-1}, x_{t:T})||p_{\theta_z}(z_t|z_{t-1})\right) \tag{4.7}$$

$$= \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_t|x_{1:T})} \log p_{\theta_x}(x_t|z_t) - \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{t-1}|x_{1:T})} \mathbb{KL}\left(q_\phi(z_t|z_{t-1}, x_{t:T})||p_{\theta_z}(z_t|z_{t-1})\right) \tag{4.8}$$

As a summary:

Deep Kalman Filter

- **generative model**

$$p_{\theta_x}(x_t|z_t) = \mathcal{N}(x_t|\mu_{\theta_x}(z_t), \text{diag}\, \sigma^2_{\theta_x}(z_t)) \tag{4.9}$$
$$p_{\theta_z}(z_t|z_{t-1}) = \mathcal{N}(z_t|\mu_{\theta_z}(z_{t-1}), \text{diag}\, \sigma^2_{\theta_z}(z_{t-1})) \tag{4.10}$$

- **inference model**

$$q_\phi(z_t|z_{t-1}, x_{t:T}) = \mathcal{N}(z_t|\mu_\phi(z_{t-1}, x_{t:T}), \text{diag}\, \sigma^2_{\theta_z}(z_{t-1}, x_{t:T})) \tag{4.11}$$

- **VLB for training**

$$\mathcal{L}(\theta, \phi, X) = \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_t|x_{1:T})} \log p_{\theta_x}(x_t|z_t) - \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{t-1}|x_{1:T})} \mathbb{KL}\left(q_\phi(z_t|z_{t-1}, x_{t:T})||p_{\theta_z}(z_t|z_{t-1})\right) \tag{4.12}$$

The $\mathbb{KL}\left(q_\phi||p_{\theta_z}\right)$'s have a close form, as the two distributions are Gaussians (see C)

From a code stand-point, following [10], we have used forward Long Short Term Memory (LSTM) to encode sequences such as $x_{1:t}$, and backward LSTM to encode sequences such as $x_{t:T}$, as inputs into the MLP parametrizing the distributions.

For example:

$$\overleftarrow{g_t} = \text{Backward LSTM}(\overleftarrow{g_{t+1}}, x_t) \text{ (encodes } x_{t:T})$$
$$q_\phi(z_t|z_{t-1}, x_{t:T}) = \mathcal{N}(z_t|\mu_\phi(z_{t-1}, \overleftarrow{g_t}), \text{diag}\, \sigma^2_\phi(z_{t-1}, \overleftarrow{g_t}))$$

The PyTorch implementation is schematized below (with B batch size, T the length of the data sequence, D_x the

dimension of the observation space, D_z the dimension of the latent space):

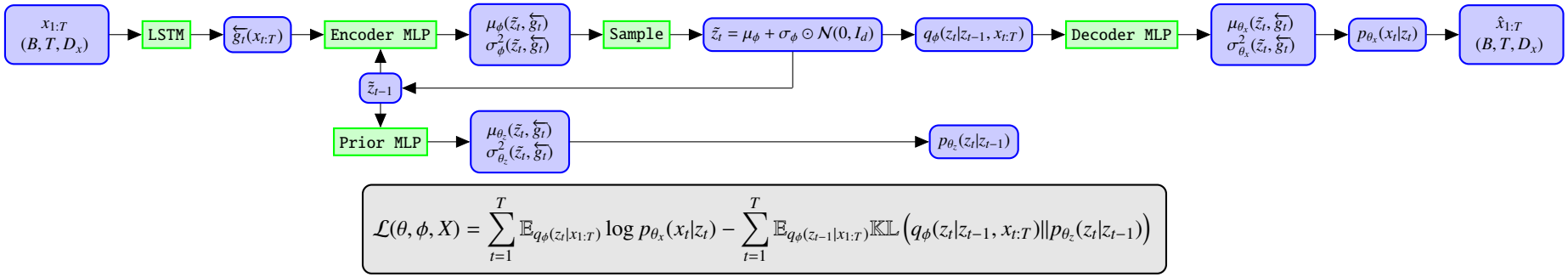


Figure 4.2: Deep Kalman Filter Model Architecture

5 Variational Recurrent Neural Network

The VRNN is the most expressive DVAE, in that sense that the general expressions 3.11, 3.12 and VLB 3.13 can not be simplified.

The GPM of the VRNN assumes full connections between latent variables, and between observed variables, to account for the full unsimplified expressions. Specifically:

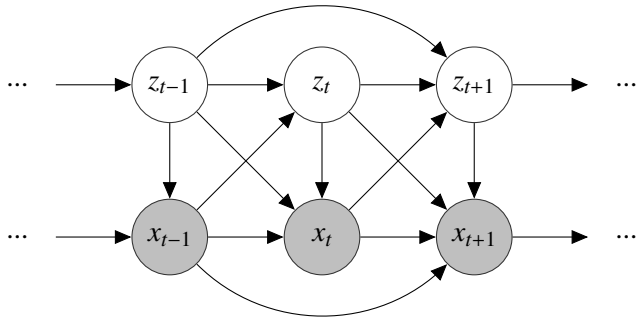


Figure 5.1: Probabilistic model of a Variational RNN

We remember that:

$$p(x_{1:T}, z_{1:T}) = \prod_{t=1}^{T} p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) p_{\theta_z}(z_t|x_{1:t-1}, z_{1:t-1})$$

And posit Gaussian distributions with diagonal covariance and mean given by two networks:

$$p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) = \mathcal{N}(x_t|\mu_{\theta_x}(x_{1:t-1}, z_{1:t}), \operatorname{diag} \sigma^2_{\theta_x}(x_{1:t-1}, z_{1:t})) \tag{5.1}$$

$$p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t-1}) = \mathcal{N}(z_t|\mu_{\theta_z}(z_{1:t-1}, x_{1:t-1}), \operatorname{diag} \sigma^2_{\theta_z}(z_{1:t-1}, x_{1:t-1})) \tag{5.2}$$

The true posterior being

$$p(z_{1:T}|x_{1:T}) = \prod_{t=1}^{T} p(z_t|z_{1:t-1}, x_{1:T})$$

we choose the encoder with the same conditional dependencies and a Gaussian expression:

$$q_\phi(z_t|z_{1:t-1}, x_{1:T}) = \mathcal{N}(z_t|\mu_\phi(z_{1:t-1}, x_{1:T}), \text{diag}\, \sigma^2_\phi(z_{1:t-1}, x_{1:T}))$$

The VLB is:

$$\mathcal{L}(\theta, \phi, X) = \sum_{t=1}^{T} \left[\mathbb{E}_{q_\phi(z_{1:t}|x_{1:T})} \log p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) - \mathbb{E}_{q_\phi(z_{1:t-1}|x_{1:T})} \mathbb{KL}\left(q_\phi(z_t|z_{1:t-1}, x_{1:T})||p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t-1})\right)\right]$$

As a summary

Variational RNN

- **generative model**

$$p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) = \mathcal{N}(x_t|\mu_{\theta_x}(x_{1:t-1}, z_{1:t}), \text{diag}\, \sigma^2_{\theta_x}(x_{1:t-1}, z_{1:t})) \tag{5.3}$$

$$p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t-1}) = \mathcal{N}(z_t|\mu_{\theta_z}(z_{1:t-1}, x_{1:t-1}), \text{diag}\, \sigma^2_{\theta_z}(z_{1:t-1}, x_{1:t-1})) \tag{5.4}$$

- **inference model**

$$q_\phi(z_t|z_{1:t-1}, x_{1:T}) = \mathcal{N}(z_t|\mu_\phi(z_{1:t-1}, x_{1:T}), \text{diag}\, \sigma^2_\phi(z_{1:t-1}, x_{1:T})) \tag{5.5}$$

- **VLB for training**

$$\begin{aligned}\mathcal{L}(\theta, \phi, X) = &\sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{1:t}|x_{1:T})} \log p_{\theta_x}(x_t|x_{1:t-1}, z_{1:t}) \\ &- \sum_{t=1}^{T} \mathbb{E}_{q_\phi(z_{1:t-1}|x_{1:T})} \mathbb{KL}\left(q_\phi(z_t|z_{1:t-1}, x_{1:T})||p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t-1})\right)\end{aligned} \tag{5.6}$$

We have chosen a different implementation from [10] and used three different LSTM networks to encode $z_{1:t}$, $x_{1:t-1}$ and $x_{t:T}$ respectively.

The PyTorch implementation is schematized below (with B batch size, T the length of the data sequence, D_x the dimension of the observation space, D_z the dimension of the latent space):

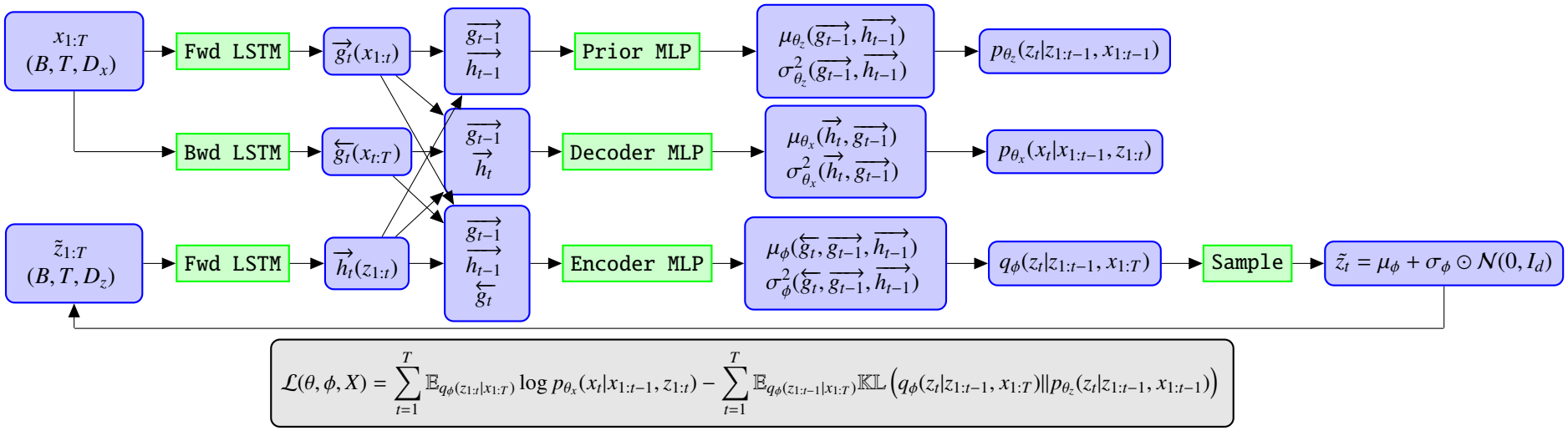


Figure 5.2: Variational RNN Model Architecture

6 Gaussian Process Variational Auto Encoder

DVAEs are a natural and straightforward extension of VAEs to the time domain. However, the discretization of time comes with limitations. First, one has to choose a relevant time interval to sample the data, which can prove arbitrary if the observed process is not well known. Second, that time interval is fixed for training and inference, can not be changed depending on the time dynamics of the observed process, and can not account for different time scales.

It is therefore interesting to allow a continuous-time formulation of the prior of the latent variables, to provide additional flexibility and expressiveness. A natural and straightforward framework for such a continuous time prior is the GP, that constitutes the core of GP-VAEs. (A summary of GP can be found in B).

If the use of GPs for time-series modeling is not recent (see for example [22] and [23]), structuring a latent prior as a GP is somewhat newer. In [2], Casale and al. build a GP-VAE generative model to predict images with different objects and views. A specific kernel is designed for the task, taking advantage of the kernel construction rules and multiplying a view-based kernel by an object-based kernel. The kernel parameters are learnt with the inference model, and the covariance matrix of the kernel is built with a low-rank approximation (VV^T) to reduce computation costs (naïvely in $O(T^3)$). In [8], Fortuin and al. focus on time-series missing values imputations. A Cauchy kernel is used, which is an instance of a Rational Quadratic Kernel, that can be viewed as an infinite sum of Gaussian kernels over the space of lengthscales. The encoder q_ϕ is a multivariate normal distribution, whose precision matrix is built mutliplying a bi-band matrix and its transpose, again to reduce computation cost. [10] cites GP-VAE but remains focused on discrete-time models. The paper [28] establishes the Markovian nature of a GP as the solution of a linear SDE, which allows to significantly reduce the computation cost of the model.

The main insight is that the solution of a linear SDE is a Gaussian Process, as the transition probabilities given by the Fokker-Plank equations are Gaussian. Thus, GP-VAEs have the potential to express naturally many phenomena described by linear SDEs. We will review later those results, issued form stochastic calculus reference books such as [16], [25] and [18].

We now move to the GP-VAE model itself.

We can consider **data taken at irregular time intervals**. We change our notation accordingly, and note $(t_1, ..., t_{i-1}, t_i, t_{i+1}, ..., t_T)$ the T times (or timestamps) considered.

The GPM of the GP-VAE is:

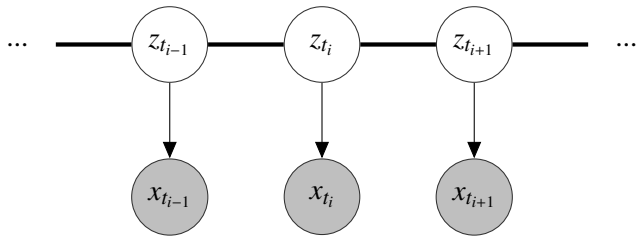


Figure 6.1: Probabilistic model of a GP-VAE

Where the **thick black line** between latent variables define a fully connected graph : all latent variables are -a priori- correlated between each other in a Gaussian Process. (NB : this GPM is not, per say, a DAG in this regard. However, D-separation still applies for observed variables x_{t_i}).

The joint distribution writes somehow differently from the one for DVAEs, as we put aside $p(z_{t_1:t_T})$:

$$p(x_{t_1:t_T}, z_{t_1:t_T}) = p(z_{t_1:t_T})p(x_{t_1:t_T}|z_{t_1:t_T}) \tag{6.1}$$

$$= p(z_{t_1:t_T}) \prod_{i=1}^{T} p(x_{t_i}|x_{t_1:t_{i-1}}, z_{t_1:t_T}) \tag{6.2}$$

$$= p(z_{t_1:t_T}) \prod_{i=1}^{T} p(x_{t_i}|z_{t_i}) \tag{6.3}$$

The prior over the latent variables $z_{t_i} \in \mathbb{R}^L$ is a set L of scalar Gaussian Process over each of the dimension $l \in \{1, ..., L\}$ of the latent variables. Formally:

$$p_{\theta_z}(z^l_{t_1:t_T}) = \mathcal{GP}(m_{\theta_z,l}(t_1 : t_T), k_{\theta_z,l}(t_1 : t_T, t_1 : t_T)) \qquad l = 1, .., L \tag{6.4}$$

where the $m_{\theta_z,l}$ are the L mean functions of the GP priors (usually chosen constant null), and the $k_{\theta_z,l}$ are the kernel functions of the GP priors.

We note at this point that:

- by design, each of the component prior of the z_{t_i} is a scalar GP, with correlation over time stamps. However, the different components of a z_{t_i} are not correlated between them.
- **The correlation across data dimensions is encoded into the observation model** $p_{\theta_x}(x_{t_i}|z_{t_i})$, whereas **the correlation in time is encoded into the latent variable GPs**.
- the kernels $k_{\theta_z,l}$ can be chosen differently to account for different prior knowledge of the data sequence. In [8] for example, Fortuin and al. uses a set of Gaussian Kernels with different lenghtscales. This provides expressiveness but also makes the models tricker to train.

Accordingly, the approximate posterior -encoder- q_ϕ is a set of L Gaussian distributions of dimension T, each one accounting for a component of z_{t_i}. Formally :

$$q_\phi(z^l_{t_1:t_T}|x^l_{t_1:t_T}) = \mathcal{N}(m^l_\phi(x_{t_1:t_T}), \Sigma^l_\phi(x_{t_1:t_T})) \qquad l = 1, .., L \tag{6.5}$$

$$= \mathcal{N}(m^l_\phi(x_{t_1:t_T}), \Lambda^l_\phi(x_{t_1:t_T})^{-1}) \tag{6.6}$$

$$= \mathcal{N}(m^l_\phi(x_{t_1:t_T}), L^l_\phi(x_{t_1:t_T})L^l_\phi(x_{t_1:t_T})^T) \tag{6.7}$$

where we have made explicit the different ways of defining the multivariate normal distribution, with its covariance matrix Σ^l_ϕ, its precision matrix Λ^l_ϕ, or with a Cholesky decomposition $L^l_\phi {L^l_\phi}^T$.

The observation model, by D-separation, is:

$$p(x_{t_1:t_T}|z_{t_1:t_T}) = \prod_{i=1}^{T} p_{\theta_x}(x_{t_i}|z_{t_i}) \tag{6.8}$$

The log-likelihood of the data writes:

$$\log p(x_{t_1:t_T}) = \log \frac{p(x_{t_1:t_T}, z_{t_1:t_T})}{p(z_{t_1:t_T}|x_{t_1:t_T})} \tag{6.9}$$

As usual, we multiply by $q_\phi(z_{t_1:t_T}|x_{t_1:t_T})$ and integrate over $dz_{t_1:t_T}$ to form the VLB:

$$\log p(x_{t_1:t_T}) = \int q_\phi(z_{t_1:t_T}|x_{t_1:t_T}) \log \frac{p(x_{t_1:t_T}, z_{t_1:t_T})}{q_\phi(z_{t_1:t_T}|x_{t_1:t_T})} \frac{q_\phi(z_{t_1:t_T}|x_{t_1:t_T})}{p(z_{t_1:t_T}|x_{t_1:t_T})} dz_{t_1:t_T} \tag{6.10}$$

$$= \mathbb{E}_{q_\phi(z_{t_1:t_T}|x_{t_1:t_T})} \log \frac{p(x_{t_1:t_T}, z_{t_1:t_T})}{q_\phi(z_{t_1:t_T}|x_{t_1:t_T})} + \mathbb{KL}\left(q_\phi(z_{t_1:t_T}|x_{t_1:t_T}) \| p(z_{t_1:t_T}|x_{t_1:t_T})\right) \tag{6.11}$$

$$\geq \mathbb{E}_{q_\phi(z_{t_1:t_T}|x_{t_1:t_T})} \log \frac{p(x_{t_1:t_T}, z_{t_1:t_T})}{q_\phi(z_{t_1:t_T}|x_{t_1:t_T})} = \mathcal{L}(\theta, \phi, X) \tag{6.12}$$

Factoring in 6.1 and 6.8, we get:

$$\mathcal{L}(\theta, \phi, X) = \mathbb{E}_{q_\phi(z_{t_1:t_T}|x_{t_1:t_T})} \log \left[\left(\prod_{i=1}^{T} p_{\theta_x}(x_{t_i}|z_{t_i}) \right) \frac{p_{\theta_z}(z_{t_1:t_T})}{q_\phi(z_{t_1:t_T}|x_{t_1:t_T})} \right] \tag{6.13}$$

$$= \sum_{i=1}^{T} \mathbb{E}_{q_\phi(z_{t_1:t_T}|x_{t_1:t_T})} \log p_{\theta_x}(x_{t_i}|z_{t_i}) - \mathbb{KL}\left(q_\phi(z_{t_1:t_T}|x_{t_1:t_T}) \| p_{\theta_z}(z_{t_1:t_T})\right) \tag{6.14}$$

We have $\mathbb{E}_{q_\phi(z_{t_1:t_T}|x_{t_1:t_T})} f(z_{t_i}) = \mathbb{E}_{q_\phi(z_{t_i}|x_{t_1:t_T})} f(z_{t_i})$ for any f, so we get finally:

$$\mathcal{L}(\theta, \phi, X) = \sum_{i=1}^{T} \mathbb{E}_{q_\phi(z_{t_i}|x_{t_1:t_T})} \log p_{\theta_x}(x_{t_i}|z_{t_i}) - \mathbb{KL}\left(q_\phi(z_{t_1:t_T}|x_{t_1:t_T}) \| p_{\theta_z}(z_{t_1:t_T})\right) \tag{6.15}$$

We note that:

- the $\mathbb{KL}$-divergence is actually the sum of the L $\mathbb{KL}$-divergences $\mathbb{KL}\left(q_\phi^l \| p_{\theta_z}^l\right)$, which have a close form solution as both distributions are Gaussian. (see the well-known result C)

- the reconstruction loss term requires sampling from $q_\phi(z_{t_i}|x_{t_1:t_T})$ using the reparameterization trick as usual.

- the GP priors $p_{\theta_z}(z_{t_1:t_T})$ depend only on the time stamps $t_1, ...t_T$. If the kernel parameters are fixed -such as in [8]- then the priors can be computed before the training loop. If the kernel parameters are learnt with the weights of the neural nets (such as in [28]), then the computation must occur at each training iteration.

As a summary:

Gaussian Process VAEs

- **generative model**

$$p(x_{t_1:t_T}, z_{t_1:t_T}) = p(z_{t_1:t_T}) \prod_{i=1}^{T} p(x_{t_i}|z_{t_i}) \tag{6.16}$$

$$p_{\theta_z}(z^l_{t_1:t_T}) = \mathcal{GP}(m_{\theta_z,l}(t_1 : t_T), k_{\theta_z,l}(t_1 : t_T)) \qquad l = 1, .., L \tag{6.17}$$

- **inference model**

$$q_\phi(z^l_{t_1:t_T}|x^l_{t_1:t_T}) = \mathcal{N}(m^l_\phi(x_{t_1:t_T}), \Sigma^l_\phi(x_{t_1:t_T})) \qquad l = 1, .., L \tag{6.18}$$

$$= \mathcal{N}(m^l_\phi(x_{t_1:t_T}), \Lambda^l_\phi(x_{t_1:t_T})^{-1}) \tag{6.19}$$

$$= \mathcal{N}(m^l_\phi(x_{t_1:t_T}), L^l_\phi(x_{t_1:t_T}) L^l_\phi(x_{t_1:t_T})^T) \tag{6.20}$$

- **VLB for training**

$$\mathcal{L}(\theta, \phi, X) = \sum_{i=1}^{T} \mathbb{E}_{q_\phi(z_{t_i}|x_{t_1:t_T})} \log p_{\theta_x}(x_{t_i}|z_{t_i}) - \mathbb{KL}\left(q_\phi(z_{t_1:t_T}|x_{t_1:t_T}) \| p_{\theta_z}(z_{t_1:t_T})\right) \tag{6.21}$$

The PyTorch implementation is schematized here:

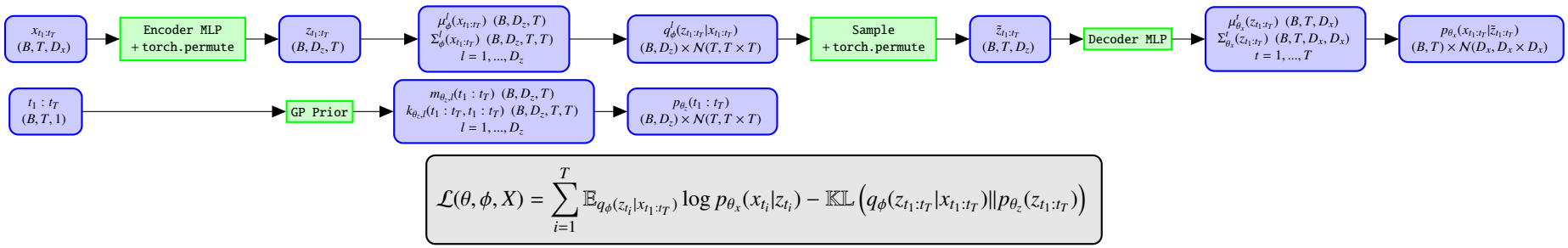


Figure 6.2: Gaussian Process VAE Model Architecture

Part III

Dynamical Variational Autoencoders - Implementation and Experiments

7 Experiments

7.1 Deep Kalman Filter

The notebook for the Deep Kalman Filter is

```
Train_DKF_Final.ipynb
```

We trained the DKF on 200 synthetic univariate time series of 100 time steps for training and 50 time steps to predict. The time series are the sum of two sine waves with noise.

```
    f1,f2,o1,o2 = np.random.rand(4, batch_size, 1)  # return 4 values for each time series
    time = np.linspace(0, 1, n_steps)  # time vector

    series = 0.4 * np.sin((time - o1) * (f1 * 5 + 10)) # first sine wave
    series += 0.2 * np.sin((time - o2) * (f2 * 20 + 20)) # second sine wave
    series += noise * (np.random.randn(batch_size, n_steps) - 0.5)  # add noise
```

The training of the DKF proved difficult, as we experienced **posterior collapse**.

Remember that the ELBO is :

$$\mathcal{L}(\theta, \phi, X) = \sum_{i=1}^{n} \mathbb{E}_{q_\phi(z_i|x_i)} \log p_{\theta_x}(x_i|z_i) - \sum_{i=1}^{n} \mathbb{KL}(q_\phi(z_i|x_i)||p_{\theta_z}(z_i))$$

The posterior collapse describes the situation where the training is such that:

- the posterior $q_\phi(z|x)$ becomes independent of x, ie $q_\phi(z|x) \approx q_\phi(z) \approx p_{\theta_z}(z)$
- the decoder is expressive enough to model the data without the latent variable: $p_{\theta_x}(x|z) \approx p_{\theta_x}(x)$

One way to avoid this is to introduce a β-scheduler, ie a varying weight between the $\mathbb{KL}$ term and the reconstruction term. First, $\beta = 0$, the total loss is exclusively the reconstruction loss until it reaches a certain threshold. Then β increases to incorporate gradually the $\mathbb{KL}$ term.

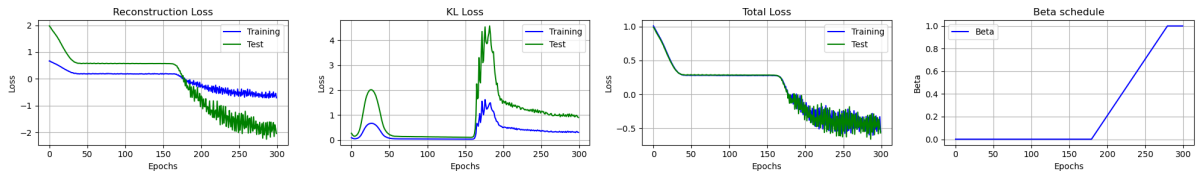


Figure 7.1: Training DKF

The model was able to reconstruct the time-series and generates plausible predictions.

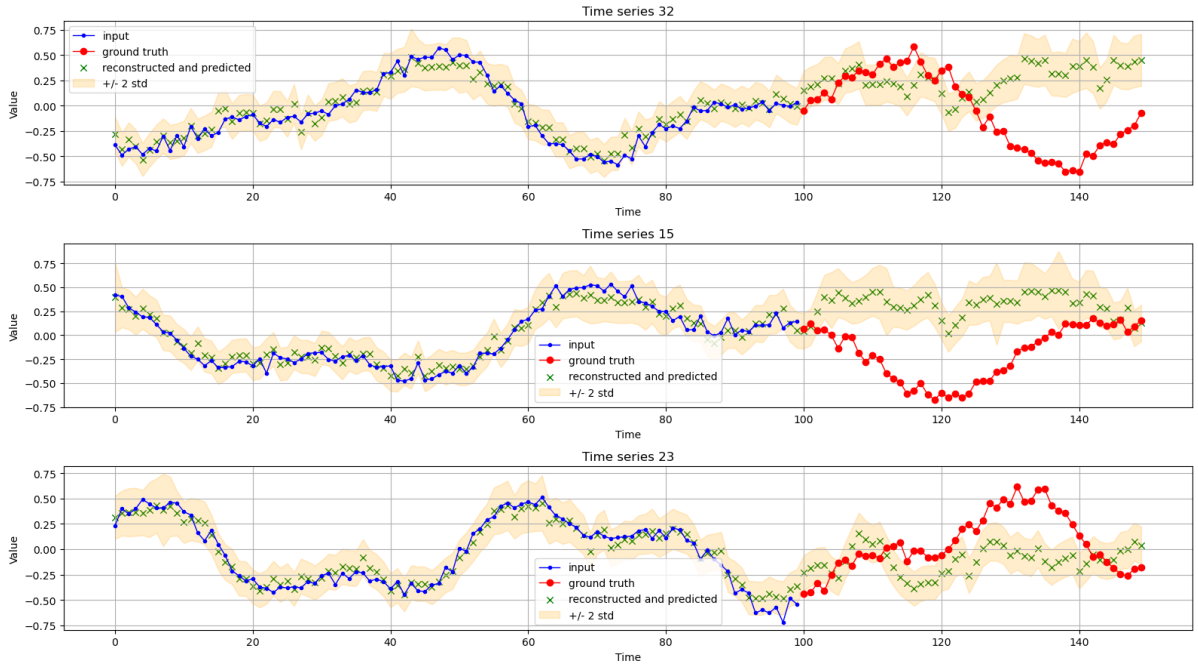


Figure 7.2: Predictions Generations DKF

7.2 Variational Recurrent Neural Network on time series

We tested a VRNN on a toy time serie dataset in

```
Train_VRNN_v1.ipynb
```

The time series are shorter than for the DKF to reduce training time - thus making the task easier. Results are promising though.

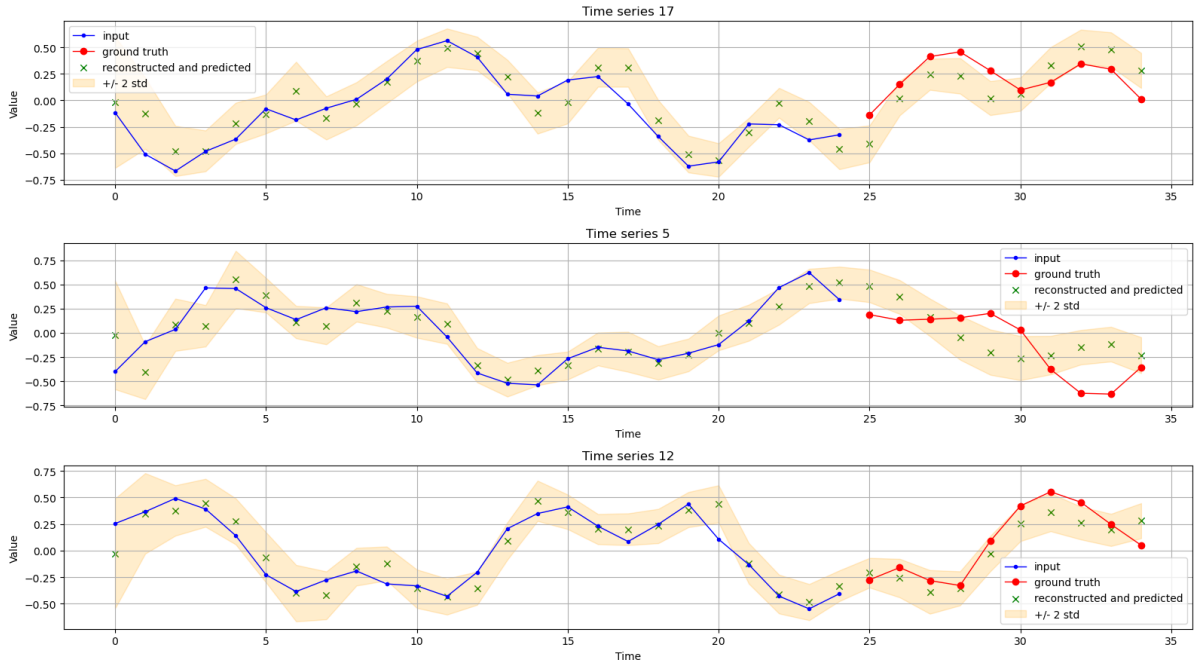


Figure 7.3: Predictions and Generations VRNN

7.3 Sprites Dataset

We will be using the **Sprites** dataset from [15].

The Sprites dataset is a synthetic cartoon character dataset of RGB images $64 \times 64 \times 3$. Each character has 4 attributes (hair, shoes, top cloth, bottom cloth) that can take 6 possibles values. Each character has three possible poses (left, front, right) and can perform 3 possible actions (walk, spell, slash) accross 8 consecutive frames.

There is a total of $6^4 \times 3 \times 3 = 11664$ series of 8 images each, that are divided between a training set and a test set.

Here is one character:

Figure 7.4: One sprite

And 5 series:

Figure 7.5: sprite series

7.4 Variational Recurrent Neural Network

We trained a VRNN in

```
Train_VRNN_Sprites_v1.ipynb
```

with a CNN encoder and a CNN decoder. The three RNNs of the VRNN have a dimension 128, and the latent space dimension is 64.

The training took a little bit more than one hour on a RTX 3090.

The reconstruction is good

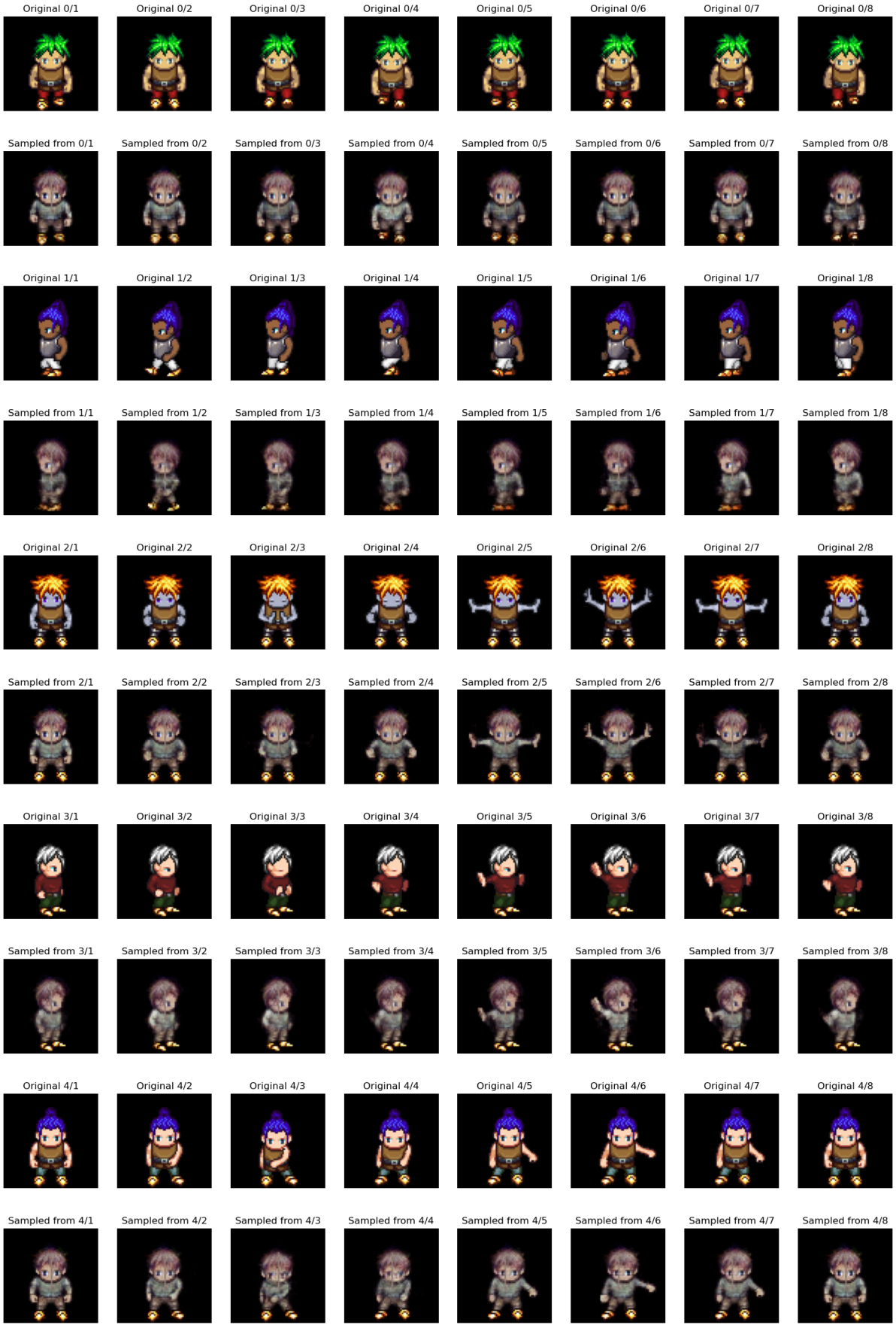


Figure 7.6: VRNN Sprites reconstruction

We tested the generation over 20 time steps, with one character as a ”seed” to intialize the first latent variable. The model can sometimes chain different motions over those 20 steps.

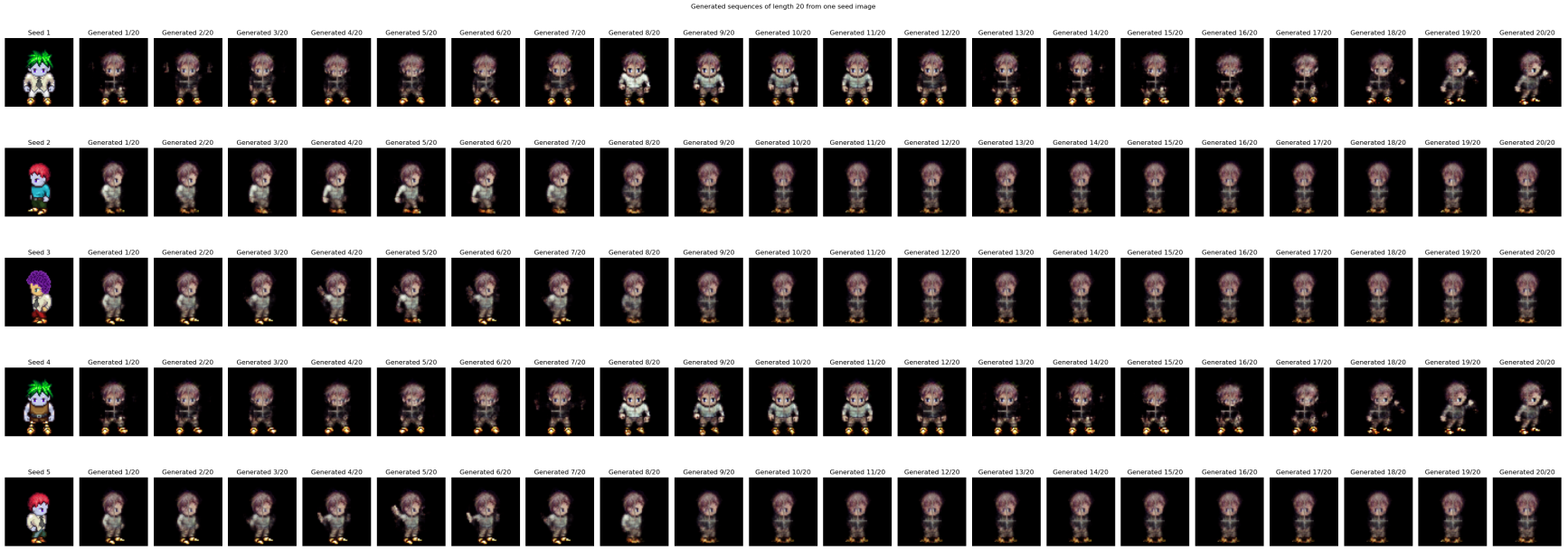


Figure 7.7: VRNN Sprites generation

7.5 Gaussian Process Variational Auto Encoder

The training of the GP-VAE proved challenging.

See the detailed training logic in: We trained a VRNN in

```
Train_GPVAE_step_by_step_logic.ipynb
```

The specific notebook for the sprite training is

```
Train_GPVAE_Sprites_XPs_v2.ipynb
```

We coded three different ways of computing a covariance matrix:

- covariance matrix
- precision matrix
- Cholesky decomposition

We coded Gaussian kernels and Matern kernels (with $\nu = 0.5, 1.5, 2.5$). The

class GaussianKernelFixed

and

class MaternKernelFixed

have fixed non-trainable lengthscale and sigma parameters (sigma is the scaling factor multiplying the kernel).

The

class GaussianKernel

and

class MaternKernel

have learnable lengthscale and sigma parameters :

```
self.lengthscale = nn.Parameter(torch.tensor(lengthscale))  # learnable lengthscale parameter
self.sigma = nn.Parameter(torch.tensor(sigma))  # learnable variance parameter
```

However, the attributes must be cloned before being used in the computations to avoid attempts to run multiple backward passes in the computation graph and trigger a RunTime Error :

```
lengthscale = self.lengthscale.clone()
sigma = self.sigma.clone()
# ...
kernel = torch.exp(-0.5 * torch.pow(torch.div(t1_b - t2_b, lengthscale),2))  # (..., N, M)
# ...
gaussian_kernel_matrix = sigma**2 * kernel
```

After several tests, we used a total of 32 GPs priors of the 4 different kernel types, with different lengthscales:

```
Dz = 32
delta_t = 1.0  # time step between two frames if T=8 frames in [0,1]

kernels_list = [ GaussianKernelFixed(lengthscale=(delta_t / (2**i)), sigma=1.0).to(device)
for i in range(int(Dz/4)) ] + \
[ MaternKernelFixed(nu=0.5, lengthscale=(delta_t / (2**i)), sigma=1.0).to(device)
for i in range(int(Dz/4)) ] +  \
```

```
[ MaternKernelFixed(nu=1.5, lengthscale=(delta_t / (2**i)), sigma=1.0).to(device)
for i in range(int(Dz/4)) ] + \
[ MaternKernelFixed(nu=2.5, lengthscale=(delta_t / (2**i)), sigma=1.0).to(device)
for i in range(int(Dz/4)) ]

mean_functions_list = [ GPNullMean().to(device) for _ in range(Dz) ] # list of Dz identical mean functions
mean, kernel_matrix, L_matrix, p_theta_z = compute_gp_priors(t, Dz, kernels_list, mean_functions_list,
verbose=True)
```

The reconstruction is good, though not perfect. We show here three samples from the reconstructed distribution of three series from the test set.

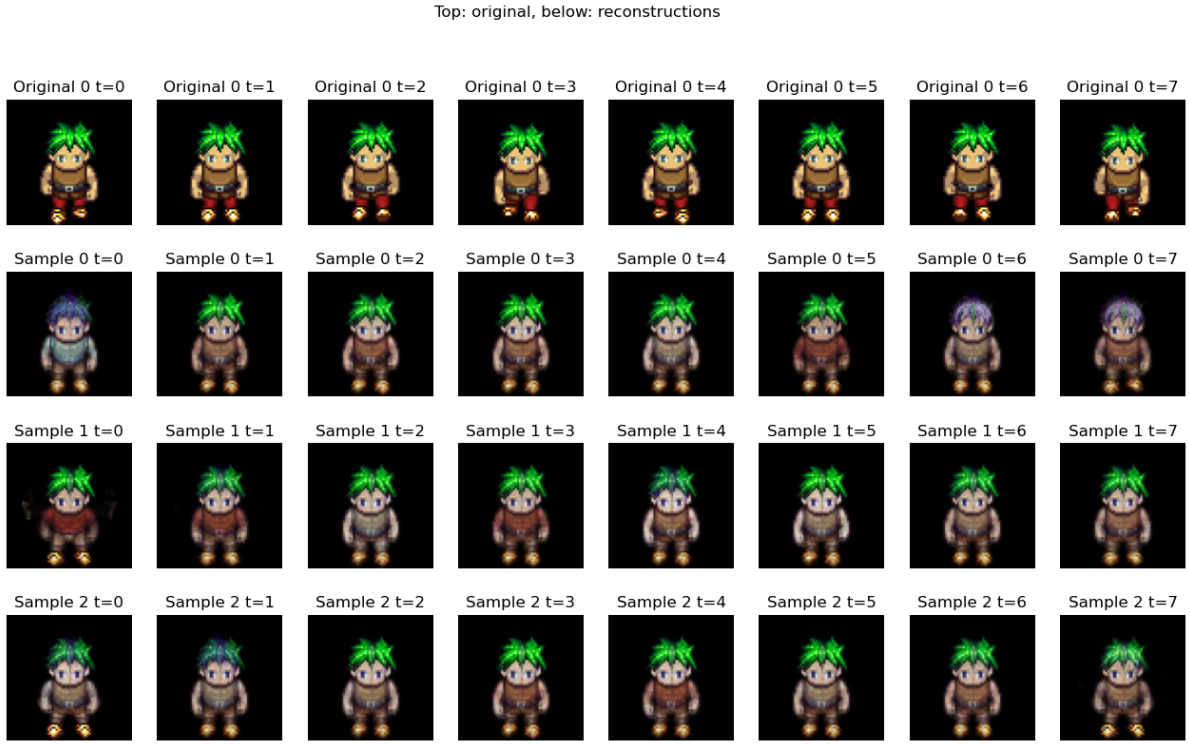


Figure 7.8: GPVAE Sprites reconstruction 1

Figure 7.9: GPVAE Sprites reconstruction 2

Figure 7.10: GPVAE Sprites reconstruction 3

The generation is not perfect yet!

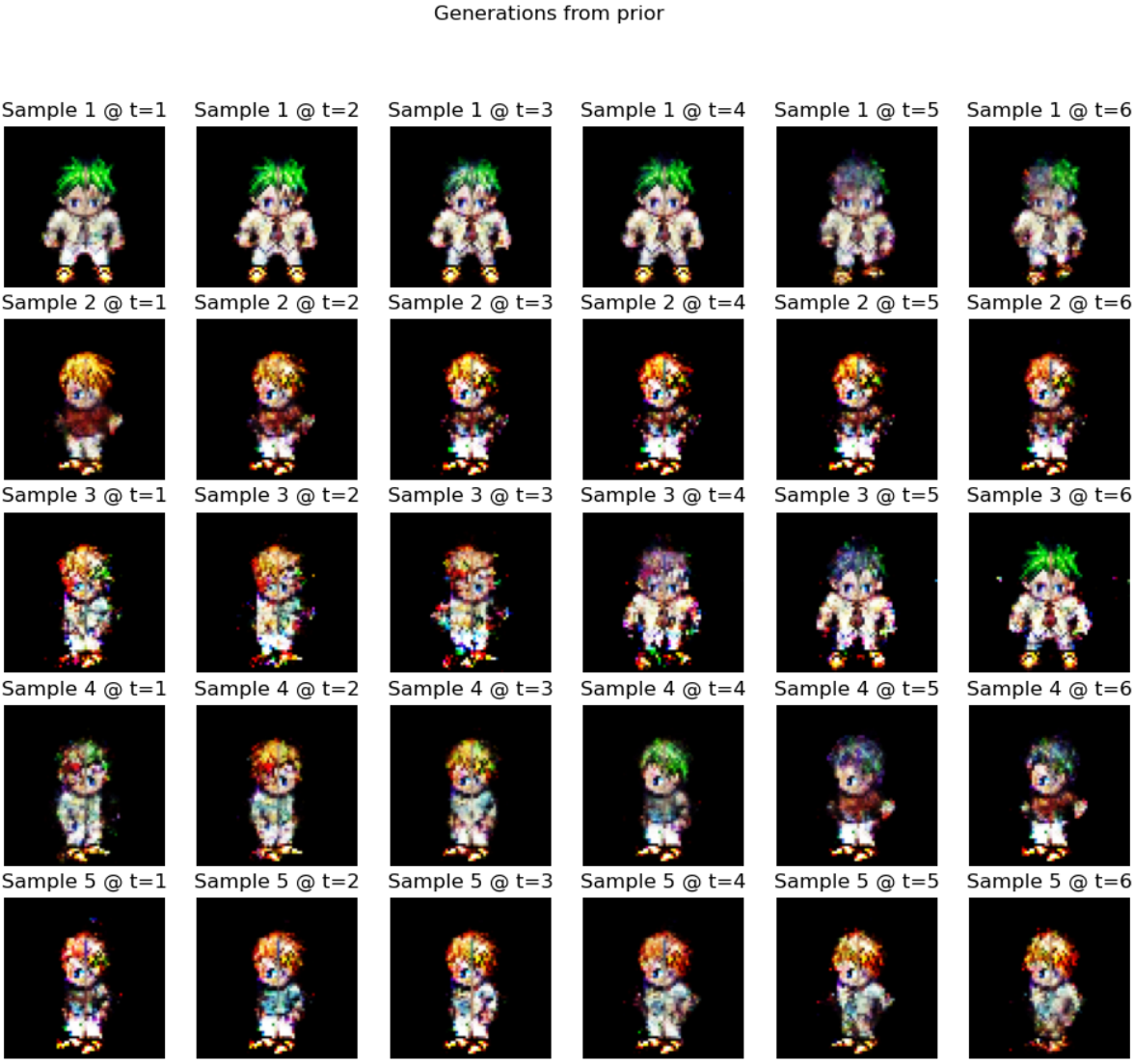


Figure 7.11: GPVAE Sprites generation 1

Part IV

Notions on stochastic differential equations and their relationships to DVAEs

This part intends to present a self-contained "survival kit" material on stochastic calculus, in order to highlight the relationships between SDEs and DVAEs.

NB : stochastic calculus is a full mathematical field in itself, and a more detailed presentation is located at the very end of this report. For a full study of the matter, the interested reader will likely enjoy [16], [25], and refer to [18].

8 Stochastic calculus intoduction

This chapter is a reminder of some key notions of stochastic calculus. More details are presented at the end of the report, and in [16], [25].

A **stochastic process** is formally defined as:

Definition 8.0.1: Stochastic process

A stochastic process X is defined as:

$$X = (\Omega, \mathcal{F}, (X_t)_{t\in T}, \mathbb{P}) \tag{8.1}$$
$$= (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in T}, (X_t)_{t\in T}, \mathbb{P}) \tag{8.2}$$

where:

- Ω is a set (universe of possibles).
- $\mathcal{F}$ is a σ-algebra of parts of Ω
- $\mathbb{P}$ is a probability measure on $(\Omega, \mathcal{F})$
- $T \subset \mathbb{R}_+$ represents time
- $(\mathcal{F}_t)_{t\in T}$ is a **filtration**, ie an increasing family of sub-σ-algebras of $\mathcal{F}$ indexed by $t : \forall 0 \leq s \leq t \in T$, $\mathcal{F}_s \subset \mathcal{F}_t \subset \mathcal{F}$.
- $(X_t)_{t\in T}$ is a family of RV defined on $(\Omega, \mathcal{F})$ with values in a measurable space $(E, \mathcal{E})$ or more simply $(E, \mathcal{B}(E))$ (set E endowed with its Borelian σ-algebra).
- $(X_t)_{t\in T}$ is assumed **adapted to the filtration** $(\mathcal{F}_t)_{t\in T}$, meaning $\forall t \in T$, X_t is $\mathcal{F}_t$-measurable

A filtration $\mathcal{F}_{t\geq 0}$ is often viewed and introduced as the *set of information available at time t*.

The core of stochastic calculus is the stochastic process known as **Brownian motion**, or **Wiener process**. We use here the definition of a multivariate Brownian motion (such as in [25]):

Definition 8.0.2: Brownian motion

A stochastic process $B = (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, (B_t)_{t\geq 0}, \mathbb{P})$ with values in $\mathbb{R}^d$ is called **Brownian motion** iff:

- $B_0 = 0$ $\mathbb{P}$-a.s.
- $\forall 0 \leq s \leq t$, the random variable $B_t - B_s$ is independent from $\mathcal{F}_s$.
- $\forall 0 \leq s \leq t$, $B_t - B_s \sim \mathcal{N}(0, Q(t-s))$
- B is continuous [a]

where the matrix $Q \in \mathbb{S}_d^{++}$ is called the **diffusion matrix**.

[a] or more exactly there exists a continuous version of B, see [16]

Meaning : the process B starts from 0, its increments are independent from the past, its increments over disjoint time intervals are independent of each other, its increments follow a centered normal law of variance equal to the length of the time interval multiplied by the diffusion matrix. NB : some authors choose the define the diffusion matrix (or scalar) outside of the Brownian motion ([16])

A core result is that the quadratic variation of the Brownian motion over an interval $[s, t]$ (equiped with a subdivison $\pi = \{s = t_0 < t_1 < ... < t_k < ... < t_n = t\}$), and defined as the limit when $|\pi| \to 0$ of $V_\pi^{(2)} = \sum_{k=0}^{n-1} |f(t_{k+1}) - f(t_k)|^2$, is:

$$\lim_{|\pi|\to 0} V_\pi^{(2)} = Q(t-s) \text{ in } L^2 \tag{8.3}$$

Or, heuristically,

$$\mathbb{E}(dB_t dB_t^T) = Qdt \tag{8.4}$$

Ito then proceeds to define **stochastic integrals**, starting with elementary processes:

Definition 8.0.3: Elementary process

A stochastic process $X = (X_s)_{s\in[a,b]}$ is called **elementary** if there exists a subdivision $a = t_0 < t_1 < ... < t_n = b$ of $[a, b]$, such that:

$$\forall t \in [a, b], \forall \omega \in \Omega, X_t(\omega) = \sum_{i=0}^{n-1} X_i(\omega) \mathbf{1}_{[t_i, t_{i+1}[}(t)$$

with $\forall i \in \{0, 1, .., n-1\}$, X_i is $\mathcal{F}_{t_i}$-measurable.
This means that, in each interval $[t_i, t_{i+1}[$, $X_t(\omega)$ is independent of t and $X_t(\omega) = X_i(\omega)$.
We define $\mathcal{E}$ (resp. $\mathcal{E}_n, n > 0$) the set of all elementary processes on $[a, b]$ (resp. the subset of the $X \in \mathcal{E}$)) such that all X_i have a finite moment $\mathbb{E}|X_i| < \infty$ (resp $\mathbb{E}(|X_i|^n) < \infty$)

Definition 8.0.4: Stochastic integral of an elementary process

Let $X \in \mathcal{E}$, ie

$$X_t(\omega) = \sum_{i=0}^{n-1} X_i(\omega) \mathbf{1}_{[t_i, t_{i+1}[}(t)$$

The stochastic integral of X is the real random variable :

$$\int_a^b X_t dB_t := \sum_{i=0}^{n-1} X_i (B_{t_{i+1}} - B_{t_i})$$

The notion is then extended to other stochastic processes (in spaces of square integrable processes, see the annex).

The definition of a SDE is derived from the notion of stochastic integral:

Definition 8.0.5: Ito's process

A process $X = (X_t)_{t \in [0,T]}$ is called a **Ito's process** if it can be written as:

$$X_t = X_0 + \int_0^t a_s ds + \int_0^t b_s dB_s \quad \forall t \in [0, T] \tag{8.5}$$

where a and b are two stochastic processes such that the integrals exist (ie $a \in \Lambda^1$ and $b \in \Lambda^2$).
Equivalently, we write X_t as the soltuion to the **Stochastic Differential Equation**:

$$dX_t = a_t dt + b_t dB_t$$

The very famous **Ito's formula** will allow to make calulations on stochastic processes:

Theorem 8.0.6: Itô's formula

An Itô's process remains an Itô's process when it is transformed by a deterministic function that is "smooth enough".
Let X be a Itô's process on $[0, T] : dX_t = a_t dt + b_t dB_t$.
Let:

$$\begin{aligned} f : \mathbb{R} \times \mathbb{R} &\to \mathbb{R} \\ (x, t) &\mapsto f(x, t) \end{aligned}$$

be $C^{2,1} : C^2$ in x, and C^1 in t.
Then $(f(X_t, t))_{t \in [0,T]}$ is also an Itô's process and:

$$d\left(f(X_t, t)\right) = \frac{\partial f}{\partial t}(X_t, t)dt + \frac{\partial f}{\partial x}(X_t, t)dX_t + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(X_t, t)b_t^2 dt \tag{8.6}$$

The last term is Itô's complementary term.
In dimension $d > 1$:

$$d\left(f(X_t, t)\right) = \frac{\partial f}{\partial t}(X_t, t)dt + (\nabla f)^T(X_t, t)dX_t + \frac{1}{2}\mathrm{Tr}\left((\nabla\nabla^T f)dX_t dX_t^T\right) \tag{8.7}$$

Proof for Theorem.

see book [16] for a clean proof. A heuristic process can be derived by using a Taylor-Lagrange decomposition at order 2, and using 8.4 ∎

9 Stochastic Differential Equations

We use here the notations of [25] and recall the key results relevant to this report.

9.1 Generic SDE

A generic **stochastic differential equation** is defined as:

Definition 9.1.1: General Form of a Stochastic Differential Equation

We define a SDE in dimension D.
Let:

- B be a Brownian motion $B_t \in \mathbb{R}^S$, of diffusion matrix Q
- F be a deterministic function "drift" $F : \mathbb{R}^D \times \mathbb{R} \to \mathbb{R}^{D \times D}$
- L be a deterministic function "dispersion" $L : \mathbb{R}^D \times \mathbb{R} \to \mathbb{R}^{D \times S}$

The SDE is:

$$dX_t = F(X_t, t)dt + L(X_t, t)dB_t \tag{9.1}$$
$$X_{t_0} = X_0 \tag{9.2}$$

where X_0 can be a scalar constant or a random variable. A stochastic process X is said to be solution of 9.1 if it verifies:

$$\forall t,\ X_t = X_0 + \int_0^t F(X_u, u)du + \int_0^t L(X_u, u)dB_u$$

As for ODE, a solution to 9.1 might not exist. Also, results similar to Cauchy-Lipschitz exist for existence and unicity, based on assumptions on F and L.

Intuitively, we can see that an "infinitesimal increment" of X_t to $X_{t+\Delta_t}$ verifies : $\Delta X_t \approx F(X_t, t)\Delta t + L(X_t, t)dB_t$. But dB_t is a Brownian increment independent of $X_{<t}$ (ie $\mathcal{F}_t$), This suggests that $X_{t+\Delta_t}$ depends on the past only by X_t. In other words, $X_t|\mathcal{F}_s = X_t|X_s$ for any $0 < s < t$. ie : **the solution of a SDE is a Markov process**. (The formal proof is given in [16].)

Formally, a Markov process is caracterized by its **transition kernels**. That is, for any $s < t$, and any $A \in \mathcal{B}_{\mathbb{R}^D}$, a Markov process verifies $\mathbb{P}(X_t \in A|\mathcal{F}_s) = \mathbb{P}(X_t \in A|X_s)$. And the transition kernels of X are the applications

$P_{s,t} : \mathbb{R}^D \times \mathcal{B}_{\mathbb{R}^D} \to [0,1]$, such that for any $f : \mathbb{R}^D \to \mathbb{R}$ measurable and bounded, we have:

$$P_{s,t}f(x) = \int_{\mathbb{R}^D} P_{s,t}(x, dy)f(y) \tag{9.3}$$

So $P_{s,t}$ actually is the probability measure of starting from x at time s, and reach $y \in dy$ at time t.

When the transition kernels have densities $p(x,t|y,s)$ (ie starting from y at time s, and reaching x at time t), then a fundamental result is the **Fokker Plank Kolmogorov** equation (also known as forward Kolomogorov) :

$$\frac{\partial p}{\partial y} = \mathcal{A}^* p \tag{9.4}$$

$$\mathcal{A}^*(\bullet) = -\sum_{i=1}^{D} \frac{\partial}{\partial x_i}(F_i(x,t)(\bullet)) + \frac{1}{2}\sum_{i,j=1}^{D} \frac{\partial^2}{\partial x_i \partial x_j}(L(x,t)QL(x,t)^T|_{i,j}(\bullet)) \tag{9.5}$$

The Fokker-Plank-Kolmogorov equation 9.4 allows to derive -ordinary- differential equations for the moments of X_t (see [25]). For the first two, defining

$$m(t) = \mathbb{E}(X_t) \tag{9.6}$$

$$P(t) = \mathbb{E}((X_t - m(t))(X_t - m(t))^T) \tag{9.7}$$

We have (NB : the expectations are taken w.r.t. x, ie the density probability $(p(x,t))$) :

$$\frac{dm}{dt} = \mathbb{E}(F(x,t)) \tag{9.8}$$

$$\frac{dP}{dt} = \mathbb{E}(F(x,t)(x - m(t))^T) + \mathbb{E}((x - m(t))F(x,t)^T) + \mathbb{E}(L(x,t)QL(x,t)^T) \tag{9.9}$$

9.2 Linear SDE

A particularly useful flavor of SDE is the linear SDE, that allows some close-form (or at least nicer) solutions:

Definition 9.2.1: Linear Stochastic Differential Equation

With the same notaions as 9.1:
The linear SDE is:

$$dX_t = F(t)X_t dt + L(t)dB_t \tag{9.10}$$

$$X_{t_0} = X_0 \tag{9.11}$$

In this case, the transition kernels family can be characterized as:

$$\Psi : \mathbb{R}^2 \to \mathbb{R}^D \tag{9.12}$$

$$\frac{\partial \Psi(\tau, t)}{\partial \tau} = F(\tau)\Psi(\tau, t) \tag{9.13}$$

$$\frac{\partial \Psi(\tau, t)}{\partial t} = -\Psi(\tau, t)F(t) \tag{9.14}$$

$$\Psi(\tau, t) = \Psi(\tau, s)\Psi(s, t) \text{ (Chapman-Kolmogorov)} \tag{9.15}$$

$$\Psi(\tau, t) = \Psi(t, \tau)^{-1} \tag{9.16}$$

$$\Psi(t, t) = I_d \tag{9.17}$$

And:

Proposition 9.2.2

the solution to 9.10 is:

$$X_t = \Psi(t, t_0)X_0 + \int_{t_0}^{t} \Psi(t, \tau)L(\tau)dB_\tau \tag{9.18}$$

$$X_{t_0} = X_0 \tag{9.19}$$

If F is constant, we find : $X_t = \exp F(t - t_0)X_0 + \int_{t_0}^{t} \exp F(t - \tau)L(\tau)dB_\tau$

We see from the form of 9.18 that, when X_0 is Gaussian, then X_t is a linear combination of independent Gaussian random variables, therefore Gaussian. ie : **the solution of a linear SDE is a Gaussian process**.

In this case, the first two moments of X_t are enough to fully describe the solution. The equations 9.8 simplify into:

$$\frac{dm}{dt} = F(t)m(t) \tag{9.20}$$

$$\frac{dP}{dt} = F(t)P(t) + P(t)F(t)^T + L(t)QL(t)^T \tag{9.21}$$

$$\text{with initial condition } X_0 \sim \mathcal{N}(m_0, P_0) \tag{9.22}$$

The transition density ($p(X_t = x(t)|X_s = x(s))$ can be found explicitely ($0 < s < t$):

$$p(x_t|x_s) = \mathcal{N}(x_t|m(t|s), P(t|s)) \tag{9.23}$$

$$m(t|s) = \Psi(t, s)x(s) \tag{9.24}$$

$$P(t|s) = \int_{s}^{t} \Psi(t, \tau)L(\tau)QL(\tau)^T\Psi(t, \tau)^T d\tau \tag{9.25}$$

Which allows to **discretize** the SDE as:

$$x_{t_{k+1}} = A_k x_{t_k} + q_k \tag{9.26}$$

$$q_k \sim \mathcal{N}(0, \Sigma_k) \tag{9.27}$$

$$A_k = \Psi(t_{k+1}, t_k) \tag{9.28}$$

$$\Sigma_k = \Sigma(t_{k+1}, t_k) = \int_{t_k}^{t_{k+1}} \Psi(t_{k+1}, \tau)L(\tau)QL(\tau)^T\Psi(t_{k+1}, \tau)^T d\tau \tag{9.29}$$

In practice, the **linearization of an SDE** is one of the techniques used to approximate SDE and allow computations:

$$dX_t = F(X_t, t)dt + L(X_t, t)dB_t \tag{9.30}$$

$$F(X_t, t) \approx F(m(t), t) + J_X F(m(t), t)(X_t - m(t)) \tag{9.31}$$

$$L(X_t, t] \approx L(m(t), t) \tag{9.32}$$

$$\frac{dm}{dt} = F(m(t), t) \tag{9.33}$$

$$\frac{dP}{dt} = J_X F(m(t), t)P(t)^T + P(t)J_X F(m(t), t)^T + L(m(t), t)QL(m(t), t)^T \tag{9.34}$$

$$\tag{9.35}$$

where $J_X F$ is the Jacobian of F w.r.t X.

A set of example calculations for the Ornstein-Uhlenbeck process is located in the appendix.

10

Filtering, Smoothing, and the GP-VAE

Equiped with the stochastic calculus basics, we see in this chapter that the filtering and smoothing tasks (ie computing posterior probabilities of the latent variables) provides a complete framework for the corresponding tasks in DVAEs.

We also see that, when a Gaussian process can be formulated as the solution of a linear SDE (ie when the kernel function verifies some properties), then the gaussian process regression problem of computing posterior probabilities can be performed by algorithms in linear time.

In this chapter, we consider Continuous-Time State Space Models (CT-SSMs) and Continuous-Discrete State Space Models (CD-SSMs). In both cases, the latent variables are defined by a (continuous) SDE. The observations can be defined by a second SDE, or by a set of discrete-time observations.

Formally, the CT-SSM is defined by:

Continuous-Time State Space model

$$dZ_t = F(Z_t, t)dt + L(Z_t, t)dB_t \tag{10.1}$$
$$dX_t = H(Z_t, t)dt + d\eta_t \tag{10.2}$$

where:

- $Z_t \in \mathbb{R}^D$ is the *state*, ie a stochastic process defining the latent variable.
- $B_t \in \mathbb{R}^S$ is a Brownian motion with diffusion matrix Q.
- $F \in \mathbb{R}^D$ and $L \in \mathbb{R}^{D \times S}$ are the usual drift and dispersion functions.
- $X_t \in \mathbb{R}^M$ is the *integrated* measurement (or observation) process.
- $H \in \mathbb{R}^M$ is the observation/measurement model.
- $\eta_t \in \mathbb{R}^S$ is a Brownian motion with diffusion matrix R.

NB : the observations are assumed to conditionnally independent of the state, and B_t, η_t are assumed independent. The observation model is equivalent to:

$$y_t = \frac{dX_t}{dt} = H(Z_t, t) + \epsilon_t \tag{10.3}$$
$$\epsilon_t = \frac{d\eta_t}{dt} \tag{10.4}$$

Formally, the CD-SSM is defined by:

Continuous-Discrete State Space model

$$dZ_t = F(Z_t, t)dt + L(Z_t, t)dB_t \tag{10.5}$$
$$x_k \sim p(x_k|z_{t_k}) \tag{10.6}$$

where:

- $Z_t \in \mathbb{R}^D$ is the *state*, ie a stochastic process defining the latent variable.
- $B_t \in \mathbb{R}^S$ is a Brownian motion with diffusion matrix Q.
- $F \in \mathbb{R}^D$ and $L \in \mathbb{R}^{D \times S}$ are the usual drift and dispersion functions.
- x_k are the observations taken at **discrete times** $(t_k)_{k=1,\dots,n}$

NB : the observations are assumed to conditionnally independent of the state.

We see that the GP-VAE is a specific CD-SSM, where the underlying latent stochastic process is actually a Gaussian process.

Also, the CT-SSM assumes a Gaussian observation model, whereas the CD-SSM allows more general observation models.

From a vocabulary stand-point, we will use indifferently *state* or *latent variable*, and *observation* or *measurement*.

10.1 Filtering and Smooting

Filtering is the problem of determining the posterior probability of the latent Z_t given the discrete measurements **up to** t, ie finding $p(Z_t|x_{1:k})$ with $t_k \leq t$. This corresponds to determining the generative transition probability $p_{\theta_z}(z_t|z_{1:t-1}, x_{1:t-1})$ in our DVAE setting.

In general, close-form solutions can be derived when the latent variables SDE is linear. In continuous time, we get the **Kalman-Bucy** filter equations, which discretize in the well-known **Kalman filter**.

Kalman-Bucy filter

$$dZ_t = F(t)Z_t dt + L(t)dB_t \tag{10.7}$$
$$dX_t = H(t)X_t dt + d\eta_t \tag{10.8}$$

where:

- $Z_t \in \mathbb{R}^D$ is the state/latent.
- $X_t \in \mathbb{R}^M$ is the observation/measurement.
- $B_t \in \mathbb{R}^S$ is a Brownian motion with diffusion matrix Q.
- $\eta_t \in \mathbb{R}^S$ is a Brownian motion with diffusion matrix R.
- $F \in \mathbb{R}^D$ and $L \in \mathbb{R}^{D\times S}$ are the usual drift/dynamic model and dispersion functions.
- $H \in \mathbb{R}^{D\times M}$ is the measurement/observation model

NB : the observations are assumed to conditionnally independent of the state. Then the Bayesian filter (Kalman-Bucy) is:

$$p(z_t|x_{<t}) = \mathcal{N}(Z_t|m_t, P_t) \tag{10.9}$$
$$K = PH(t)^T R^{-1} \tag{10.10}$$
$$dm = F(t)mdt + K(dX_t - H(t)mdt) \tag{10.11}$$
$$\frac{dP}{dt} = F(t)P + PF(t)^T + L(t)QL(t)^T - KRK^T \tag{10.12}$$

In practice, one can approximate a general SDE by a linear SDE and apply Kalman-Bucy.

Smoothing is the problem of determining the posterior probability of the latent Z_t given all known observations, ie finding $p(Z_t|x_{1:T})$ for all $t \in [0, T]$. This corresponds to determining the inference model $q_\phi(z_t|z_{1:t-1}, x_{1:T})$ in the DVAE setting.

We describe here briefly the RTS smoother for discrete time models.

Discretizing the transition density in CD-SSM, we have

$$Z_{t_{k+1}} \sim p(Z_{t_{k+1}}|Z_{t_k}) \tag{10.13}$$
$$X_k \sim p(X_k|Z_{t_k}) \tag{10.14}$$

And the smoothers:

Bayesian smoother

$$Z_{t_{k+1}} \sim p(Z_{t_{k+1}}|Z_{t_k}) \tag{10.15}$$
$$X_k \sim p(X_k|Z_{t_k}) \tag{10.16}$$

The *Bayesian smoother* is, for any $k < T$:

$$p(Z_{t_{k+1}}|X_{1:k}) = \int p(Z_{t_{k+1}}|Z_{t_k})p(Z_{t_k}|X_{1:k})dZ_{t_k} \tag{10.17}$$
$$p(Z_{t_k}|X_{1:T}) = p(Z_{t_k}|X_{1:k}) \int \left(\frac{p(Z_{t_{k+1}}|Z_{t_k})p(Z_{t_{k+1}}|X_{1:T})}{p(Z_{t_{k+1}}|X_{1:k}} dZ_{t_{k+1}} \right) \tag{10.18}$$

The backward recursion is started from the final step, where the filtering and smoothing densities are the same : $p(Z_{t_T}|X_{1:T})$.

The RTS smoother is the close-form solution of the Bayesian filter for a linear Gaussian problem - see [25] for the algorithm.

10.2 GP-VAE

We wrap up here linking the filtering/smoothing theory of linear SDE with the GP-VAE model of [8].

If we use the formalization above, a GP-VAE is basically:

$$Z_t \sim \mathcal{GP}(m(\bullet), k(\bullet, \bullet)) \tag{10.19}$$
$$X_{t_k} \sim p(X_{t_k}|Z_{t_k}) \tag{10.20}$$

If we assume that the observation model is Gaussian, then we get

$$Z_t \sim \mathcal{GP}(m(\bullet), k(\bullet, \bullet)) \tag{10.21}$$
$$X_{t_k} \sim \mathcal{N}(X_{t_k}|Z_{t_k}, \sigma^2) \tag{10.22}$$

Computing the posterior distribution $p(Z_t|X_{t_1:t_T})$ is performing a Gaussian Process regression (see [22]), which naively scales in $O(n^3)$.

However, if the Gaussian process can be written as a linear SDE:

$$dZ_t = F(t)Z_t dt + L(t)dB_t \tag{10.23}$$
$$X_{t_k} \sim \mathcal{N}(X_{t_k}|Z_{t_k}, \sigma^2) \tag{10.24}$$

then the Kalman filter and smoother apply, that scale in $O(n)$. This is the main idea in [28].

However - the solutions of linear SDE are Gaussian Processes, but the converse is not true. More specifically, some kernel functions are such that the associated GP can not be represented as the solution of a linear SDE.

For a given kernel function $k(t, t')$, [25] aims at finding a linear time-invariant model

$$dZ_t = FZ_t dt + LdB_t \tag{10.25}$$

$$X_{t_k} \sim \mathcal{N}(X_{t_k}|HZ_{t_k}, \sigma^2) \tag{10.26}$$

with $Z_t \in \mathbb{R}^D$, but $X_t \in \mathbb{R}$ is one-dimensional (ie $H \in \mathbb{R}^{1\times D}$), and such that $z_t = HZ_t$ is a Gaussian process with kernel k.

We give now some examples, and counter-examples, of such associations.

- **Brownian motion** : the Brownian motion is the solution of $dZ_t = dB_t$, and a GP with kernel $k(t, t') = \min(t, t')$ (see E)
- **Ornstein Uhlenbeck** : the O.U. process

$$dZ_t = -\frac{1}{l}Z_t dt + dB_t \tag{10.27}$$

  where dB_t has diffusion coefficient $\frac{2\sigma^2}{l}$, is a GP with kernel:

$$k_{\text{exp}} = \sigma^2 \exp\left(-\frac{|t-t'|}{l}\right) \tag{10.28}$$

- **Matern** : the SDE representation with

$$F = \begin{pmatrix} 0 & 1 \\ -\lambda^2 & -2\lambda \end{pmatrix} \tag{10.29}$$

$$L = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{10.30}$$

$$H = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{10.31}$$

  is a GP with the Matern kernel with $\nu = \frac{3}{2}$:

$$k_{\text{Matern}} = \sigma^2\left(1 + \frac{\sqrt{3}|t-t'|}{l}\right)\exp\left(-\frac{\sqrt{3}|t-t'|}{l}\right) \tag{10.32}$$

  and $\lambda = \frac{\sqrt{3}}{l}$, diffusion is $q = 4\lambda^3\sigma^2$.

Conversely, the following kernels can not be used to derive an associated linear SDE:

- **squared exponential** : the widely used

$$k_{\text{se}}(t, t') = \sigma^2 \exp\left(-\frac{|t-t'|^2}{2l^2}\right) \tag{10.33}$$

- **rational quadratic**:

$$k_{\text{rq}}(t, t') = \sigma^2\left(1 + \frac{|t-t'|^2}{2\alpha l^2}\right)^{-\alpha} \tag{10.34}$$

with $\alpha > 0$.

In that latter case, one can use spectral decomposition (ie Mercer's theorem, see MVA kernel class [5]) to approximate the kernel function and determine an associated linear SDE.

Part V

Conclusion

11

Conclusions and perspectives

We have seen that VAEs can be applied to data sequences by assuming a temporal relationship in the prior over latent variables. This is the framework of DVAEs. When the prior over the latent variables is discrete, this formulation produces, for example, DKF and VRNN, which we detailed, coded and tested on two toy datasets.

Assuming a continuous-time prior over the latent variables provides more expressiveness, as it can handle irregularly sampled data. A first model is GP-VAE in which the latent prior is a set of GPs over the time dimensions. We detailed, coded and tested the GP-VAE on the Sprites dataset. If this model is potentially more expressive than its discrete counterparts, it is also tricker to train, as the choice of prior reverts to a choice of kernel functions.

Factoring in stochastic calculus opens a broader perspective and a new field of models. GPs include stochastic processes solutions to linear SDEs and allow to use linear-time filtering and smoothing algorithms to reduce computation times.

The Markovian nature of stochastic processes solutions of general SDEs allows to use SDEs to formulate latent priors that are suitable for filtering and smoothing. This idea of Neural-SDEs builds on Neural-ODEs and points to a set of potentially very expressive models.

It's been quite a personal journey... and it continues.

Part VI

Complements

12

Stochastic processes main theory

We posit the following in the rest of the doc, unless specified otherwise:

Assumption 12.0.1: Main assumptions

1. $(\Omega, \mathcal{F}, \mathbb{P})$ is a probabilistic space, with Ω the universe, $\mathcal{F}$ the tribe (σ-algebra) of events, $\mathbb{P}$ a probability measure on $(\Omega, \mathbb{F})$.
2. $(E, \mathbb{B}(E))$ is a measurable space, endowed with its Borelian σ-algebra. E will sometimes be referred to as the *state space*. It is typically the space where random variables will live, most of the time $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$.
3. T is the set of *times*, typically $T = [0, a]$ with $a > 0$, or $T = [0, +\infty[$.

We view a stochastic process as a two-variable function from $T \times \Omega$ to $(E, \mathcal{B}(E))$:

$$X : T \times \Omega \longrightarrow (E, \mathcal{B}(E)) \tag{12.1}$$
$$(t, \omega) \longmapsto X(t, \omega) \tag{12.2}$$

This leads to two complementary views:

Stochastic process as a collection of random variables indexed by time We view X as a collection of random variables $(X_t)_{t \in T}$:

$$\forall t \in T, X_t : \omega \mapsto X_t(\omega) \tag{12.3}$$

1. Each random variable X_t is defined on $(\Omega, \mathcal{F}, \mathbb{P}$
2. In practice, only some events $A \in \mathcal{F}$ can occur during the times $[0, t] \in T$. We will note $\mathcal{F}_t$ the smallest σ-algebra that contains the set of events A that can occur during $[0, t]$.
3. As a result, X_t must be $\mathcal{F}_t$-measurable.

An alternate view arises when we fix ω and allow $t \in T$:

Stochastic process as a set of random trajectories We view X as

$$X : \Omega \longrightarrow \mathbb{R}^T \tag{12.4}$$
$$\omega \longmapsto X(\omega) : t \mapsto X(t, \omega) \tag{12.5}$$

1. $\forall \omega \in \Omega$, $X(\omega) \in \mathbb{R}^T$ is a random trajectory from T into $\mathbb{R}$

2. $\mathbb{R}^T$ should be endowed with an appropriate σ-algebra.

Those intuitions lead to a formal definition :

Definition 12.0.2: Stochastic process

A stochastic process **X** is defined as:

$$X = (\Omega, \mathcal{F}, (X_t)_{t\in T}, \mathbb{P}) \tag{12.6}$$
$$= (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in T}, (X_t)_{t\in T}, \mathbb{P}) \tag{12.7}$$

where:
- Ω is a set (universe of possibles).
- $\mathcal{F}$ is a σ-algebra of parts of Ω
- $\mathbb{P}$ is a probability measure on $(\Omega, \mathcal{F})$
- $T \subset \mathbb{R}_+$ represents time
- $(\mathcal{F}_t)_{t\in T}$ is a **filtration**, ie an increasing family of sub-σ-algebras of $\mathcal{F}$ indexed by t : $\forall 0 \leq s \leq t \in T$, $\mathcal{F}_s \subset \mathcal{F}_t \subset \mathcal{F}$.
- $(X_t)_{t\in T}$ is a family of RV defined on $(\Omega, \mathcal{F})$ with values in a measurable space $(E, \mathcal{E})$ or more simply $(E, \mathcal{B}(E))$ (set E endowed with its Borelian σ-algebra).
- $(X_t)_{t\in T}$ is assumed **adapted to the filtration** $(\mathcal{F}_t)_{t\in T}$, meaning $\forall t \in T$, X_t is $\mathcal{F}_t$-measurable

Often, we will have $(E, \mathcal{B}(E)) = (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, and $T = [0, a]$ or $T = [0, +\infty[$ or $T = \mathbb{N}$.

Other important notions:

- **σ-algebra generated by a random variable**. Let $X : (\Omega, \mathcal{F}, \mathbb{P}) \to (E, \mathcal{E})$ a random variable. The σ-algebra generated by X is the smallest σ-algebra that makes X-measurable. It is the set of "pre-images" of $\mathcal{B}(E)$ by X. Formally;

$$\sigma(X) = \{A = X^{-1}(B), B \in \mathcal{E}\} \tag{12.8}$$

- **natural filtration of** $(X_t)_{t\in T}$. The set of $\mathcal{F}_t = \sigma(X_s, s \leq t), \forall t \in T$, ie the set of σ-algebras generated by the RV X_s for $s \leq t$, is called the natural filtration of (X_t)

Definition 12.0.3: Finite dimension laws of a stochastic process

Let X be a stochastic process, $I = \{t_1, t_2, ..., t_n\}, t_1 < t_2 < ... < t_n$ a finite part of T, and $X_I = (X_{t_1}, X_{t_2}, ..., X_{t_n}) \in E^I$ a random vector.
Then, the **law of the random vector** X_I is the probability measure μ_I image of $\mathbb{P}$ by $X_I : \Omega \to (E^I, \mathcal{B}(E^I))$

We also remind of Gaussian processes -with a view different from [22].

Definition 12.0.4: Gaussian vector

Let $\xi = (X_1, X_2, ..., X_n)$ be a **centered** random vector in $\mathbb{R}^n$ ($\mathbb{E}(\xi) = 0$), verifying $\xi \in L^2(\Omega, \mathcal{F}, \mathbb{P})$ (ie $\forall i, \mathbb{E}(X_i^2) < \infty$). Let $\Gamma = \mathbb{E}(X_i X_j)|_{i,j}$ its covariance matrix (definite positive).
ξ **is a Gaussian vector** iff, equivalently:

1. $\forall a_1, a_2, ..., a_n \in \mathbb{R}^n$, $\sum_{i=1}^{n} a_i X_i$ follows a normal law (ie is Gaussian)
2. or : the characteristic function of ξ, $\Phi_\xi(t) = \mathbb{E}(e^{i<\xi|t>})$, can be written

$$\Phi_\xi(t) = e^{-\frac{1}{2}<t|\Gamma t>} \ (t \in \mathbb{R}^n)$$

With notation $\xi \sim \mathcal{N}(0, \Gamma)$.
More generally,

$$Z \sim \mathcal{N}(m, \Gamma) \iff \Phi_Z(t) = e^{i<m|t>} e^{-\frac{1}{2}<t|\Gamma t>}$$

Definition 12.0.5: Gaussian Process

Let X be a stochastic process taking its values in $E = \mathbb{R}^n$.
X **is said to be a Gaussian process iff all its finite dimension laws are Gaussian**.
If $X = (X_t)_{t \in T}$ is a real-valued Gaussian process ($E = \mathbb{R}$), we define $\forall s, t \in T$:

- $m(t) = \mathbb{E}(X_t)$ is the **mean of the Gaussian process**
- $\Gamma(t, s) = \mathbb{E}((X_t - m(t))(X_s - m(s))$ is the **covariance of the Gaussian process**

Reciprocally,

Theorem 12.0.6: Existence of a Gaussian process

Let $m : T \to \mathbb{R}$, and $\Gamma : T \times T \to \mathbb{R}$ be two functions such that:

$$\forall I = \{t_1, t_2, ..., t_n\} \text{ finite part of T } \Gamma_I = \Gamma(t_i, t_j)|_{i,j} \text{ is } \textbf{symetric definite positive}$$

then there exists a Gaussian process $X = (X_t)_{t \in T}$, that is unique at a near equivalence, such that:

$$\forall I = \{t_1, t_2, ..., t_n\} \subset T, X_I = \{X_{t_1}, X_{t_2}, ..., X_{t_n}\} \sim \mathcal{N}(m_I, \Gamma_I),$$
$$m_I = (m(t_1), m(t_2), ..., m(t_n))$$

Proof for Theorem.

p19 ∎

We will not cover notion such as stopping times or martingales, even they are central in stochastic calculus. Please refer to [16] for a detailed presentation.

The foundational stochastic process for stochastic calculus is the **brownian motion**:

Definition 12.0.7: Brownian motion

A real-valued stochastic process $B = (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, (B_t)_{t\geq 0}, \mathbb{P})$ is called **Brownian motion** iff:

- $B_0 = 0$ $\mathbb{P}$-a.s.
- $\forall 0 \leq s \leq t$, the random variable $B_t - B_s$ is independent from $\mathcal{F}_t$.
- $\forall 0 \leq s \leq t$, $B_t - B_s \sim \mathcal{N}(0, t-s)$

Meaning : the process B starts from 0, its increments are independent from the past, and follow a centered normal law of variance equal to the length of the time interval.

When $(\mathcal{F}_t)_{t\geq 0}$ is the natural filtration of $(B_t)_{t\geq 0}$, B is said to be a **natural Brownian motion**

A fundamental property of the Brownian motion is the following:

Theorem 12.0.8: Gaussian characterization of the Brownian motion

1. Let $B = (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, (B_t)_{t\geq 0}, \mathbb{P})$ be a Brownian motion. Then B verifies:
   - $B_0 = 0$ $\mathbb{P}$-a.s.
   - $\forall 0 \leq t_1 < t_2 < .. < t_n$, $(B_{t_1}, B_{t_2}, ..., B_{t_n})$ is a centered Gaussian vector.
   - $\forall s, t \geq 0$, $\mathbb{E}(B_s B_t) = \min(s, t)$

   This means that B is a real centered Gaussian process, of covariance function $\Gamma(t, s) = \min(s, t)$
2. Conversely, if B verifies the three properties above, then $(\Omega, \mathcal{F}, (\tilde{\mathcal{F}}_t)_{t\geq 0}, (B_t)_{t\geq 0}, \mathbb{P})$ is a natural Brownian motion (with $(\tilde{\mathcal{F}}_t)_{t\geq 0}$ the natural filtration of the family $(B_t)_{t\geq 0}$).

Proof for Theorem.

see [16] ∎

A second fundamental property of the Brownian motion is that its **quadratic variation** is non-zero. More formally,

Definition 12.0.9: Variations of a function

Let $f : [a, b] \to \mathbb{R}$ a function (with $a, b \in \mathbb{R}$). Let $\pi = \{a = t_0 < t_1 < ... < t_n = b\}$ a subdivision of $[a, b]$.

1. The **variation of** f **along** π is $V_\pi = \sum_{k=0}^{n-1} |f(t_{k+1} - f(t_k)|$
2. The **total variation of** f **on** $[a, b]$ is $V^\pi_{[a,b]} = \sup_\pi V_\pi$. f is said to have a **bounded variation** if $V^\pi_{[a,b]} < \infty$.
3. The **quadratic variation of** f **along** π is $V^{(2)}_\pi = \sum_{k=0}^{n-1} |f(t_{k+1}) - f(t_k)|^2$
4. The **total quadratic variation of** f **along** $[a, b]$ is :

$$[f]_{a,b} = \lim_{|\pi|\to 0} V^{(2)}_\pi$$
$$\text{with } |\pi| = \max_k |t_{k+1} - t_k|$$

The total quadratic variation has a slightly different definition from the total variation.

Theorem 12.0.10: Quadratic variation of a Brownian motion

Let $B = (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, (B_t)_{t\geq 0}, \mathbb{P})$ a Brownian motion. Then $\forall\, 0 < s \leq t$

$$\lim_{|\pi|\to 0} V_\pi^{(2)} = t - s \text{ in } L^2 \tag{12.9}$$

$$\lim_{|\pi|\to 0} \mathbb{E}\left(|V_\pi^{(2)} - (t-s)|^2\right) = 0 \tag{12.10}$$

Proof for Theorem.

see [16] ∎

The reader will refer to [16] for proofs of existence, continuity, and nowhere-differentiability of the Brownian motion.

Last, we introduce the notion of **Markov Process**, which is a generalization of the Markov Chains to continuous time.

Intuitively, building on the discrete-time Markov chain, a **Markov Process** is a stochastic process in which the behavior at time t given the information available up to time s (with $0 < s < t$) depends only on the most recent past, ie the information at time s only.

In other words, for $s < t, \in T$, the law of $X_t|\mathcal{F}_s$ depends only on X_s.

The intuition is then:

Assumption 12.0.11: Intuition of a Markov Process

The stochastic process X is called a **Markov Process** if:
$\forall s, t \in T,\ s < t,\ \forall A \in \mathcal{B}_E,\ \mathbb{P}(X_t \in A|\mathcal{F}_s) = \mathbb{P}(X_t \in A|X_s)$

We note **the transition probability of, given a start from x at time s, to reach $A \in \mathcal{B}_E$ at time t**;

$$\mathbb{P}(X_t \in A|X_s = x) = P_{s,t}(x, A) \tag{12.11}$$

We see that $A \mapsto P_{s,t}(x, A)$ is a **probability measure** on $\mathcal{B}_E$, that we note:

$$P_{s,t}(x, dy) : \mathcal{B}_E \to [0, 1] \tag{12.12}$$

$$A \mapsto P_{s,t}(x, A) \tag{12.13}$$

We now consider the space $C = \{f : E \to \mathbb{R}, \text{borelian, bounded}\}$, and the operator $P_{s,t}$ from C into C defined by:

$$P_{s,t} : C \to C \tag{12.14}$$

$$f \mapsto P_{s,t}f : x \mapsto P_{s,t}f(x) = \int_E P_{s,t}(x, dy)f(y) \tag{12.15}$$

$$P_{s,t}f(x) = \int_E f(y)P_{s,t}(x, dy) = \mathbb{E}(f(X_t)|X_s = x) \tag{12.16}$$

Taking $f = \mathbb{1}_A$, we recover 12.11 from 12.16.

We are now equipped to define:

Definition 12.0.12: Transition kernels

A family $(P_{s,t})_{s<t,\in T}$ of applications $(E, \mathcal{B}_E) \to [0,1]$ is said to be a family of **transition kernels** iff:

- $\forall s < t, \forall A \in \mathcal{B}_E, P_{s,t}(\bullet, A) : x \mapsto P_{s,t}(x, A)$ is measurable.
- $\forall s < t, \forall x \in E, P_{s,t}(x, \bullet) : A \mapsto P_{s,t}(x, A)$ is a probability measure on $\mathcal{B}_E$.
- The **Chapman Kolmogorov** property holds:

$$\forall x \in E, \forall A \in \mathcal{B}_E, \forall s < t < u, P_{s,u}(x, A) = \int_E P_{s,t}(x, dy) P_{t,u}(y, A) \tag{12.17}$$

That is : we start from x at time s, we arrive at a random y (with some probability distribution over y) at some intermediate time t, and then we start from y to reach A at time u.
Considering $P_{s,t}$ as operators (12.16), then Chapman-Kolmogorov writes:

$$P_{s,u} = P_{s,t} P_{t,u} \tag{12.18}$$

And define a **Markov Process**:

Definition 12.0.13: Markov Process

A stochastic process X is said to be a Markov Process with transition kernels $\{P_{s,t}; s, t \in T, s < t\}$ iff $\forall f : E \to \mathbb{R}$ Borelian and bounded, and $\forall s < t \in T$ we have:

$$\mathbb{E}(f(X_t)|\mathcal{F}_s) = P_{s,t} f(X_s) \ \ \mathbb{P} - \text{a.s} \tag{12.19}$$

The transition kernels $P_{s,t}$ are also called transition probabilities.
The law of X_0 is a probability measure ν over $\mathcal{B}_E$ defined by:

$$\nu(A) = \mathbb{P}(X_0 \in A) \tag{12.20}$$

and called **initial law of the process**.
Again, if we take $f = \mathbb{1}_A$, we recover $\mathbb{P}(X_t \in A|\mathcal{F}_s) = \mathbb{P}(X_t \in A|X_s)$.

Definition 12.0.14: Homogeneous Markov Process

A Markov Process X is said to be **homogeneous** if its transition kernels family $P_{s,t}$ depends only on $t - s$. ie:

$$\forall s < t, P_{s,t} = P_{0,t-s} := P_u \ (u = t - s). \tag{12.21}$$

We can now compute the finite dimension laws of a Markov Process:

Theorem 12.0.15: Finite dimension laws of a Markov Process

Let X be a Markov Process of initial law ν and probability transitions $P_{s,t}$. For all finite sequence of times $0 = t_0 < t_1 < ... < t_k$, and for all set of borelian bounded functions $f_i : E \to \mathbb{R}, 0 \leq i \leq k$, we have:

$$\mathbb{E}(f_0(X_0)f_1(X_1)...f_k(X_{t_k})) = \tag{12.22}$$

$$\int_E \nu(dx_0)f(x_0) \int_E P_{0,t_1}(x_0, dx_1)f_1(x_1)... \int_E P_{t_{k-1},t_k}(x_{k-1}, dx_k)f_k(x_k) \tag{12.23}$$

And $\forall A_0, A_1, ..., A_k \in \mathcal{B}_E$, with $f_i = \mathbb{1}_{A_i}$:

$$\mathbb{P}(X_0 \in A_0, X_1 \in A_1, ..., X_k \in A_k) = \int_{A_0} \nu(dx_0) \int_{A_1} P_{0,t_1}(x_0, dx_1)... \int_{A_k} P_{t_{k-1},t_k}(x_{k-1}, dx_k) \tag{12.24}$$

Proof for Theorem.

see [16] p92 ∎

Let $(C_o(E), \|\cdot\|_\infty)$ be the Banach space [1] of functions $f : E \to \mathbb{R}$ continuous, s.t. $f \underset{\infty}{\to} 0$.
Let $(P_t)_{t\geq 0}$ be a family of positive operators [2].

Definition 12.0.16: Feller semi-group

$(P_t)_{t\geq 0}$ is said to be a **Feller semi-group** if:

- $P_0 = I_d$ and $\forall t \geq 0, \|P_t\| \leq 1$
- $\forall t, t' \geq 0, P_t P_{t'} = P_{t+t'}$
- $\forall f \in C_o(E), \lim_{t\downarrow 0} \|P_t f - f\|_\infty = 0$

An homogeneous Markov process on E is said process of Feller if its semi-group is of Feller.

Definition 12.0.17: Infinitesimal Generator

X a process of Feller. Let $f \in C_o(E)$ be such that the limit below exists in $C_o(E)$:

$$\lim_{t\downarrow 0} \frac{1}{t}(P_t f - f) = Af \tag{12.25}$$

then f is said to be in the domain D_A of A operator defined by 12.25.
A is called infinitesimal generator of the semi-group $(P_t)_{t\geq 0}$.
Then we can write:

$$\mathbb{E}(f(X_{s+h}|\mathcal{F}_s) = f(X_s) + hAf(X_s) + o(h) \tag{12.26}$$

where $o(h)$ depends only on f.

Regarding the Brownian motion, we have:

[1] Banach space : complete normed vector space
[2] $f \geq 0 \implies P_t f \geq 0$

Theorem 12.0.18: Semi-group of the Brownian motion

Let $B = (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\in T}, (B_t)_{t\in T}, \mathbb{P})$ be a Brownian motion on $\mathbb{R}$. Then **B is an homogeneous Markov Process on** $\mathbb{R}$, of initial law $\nu = \delta_0$, and whose semi-group is given by (for any $f : E \to \mathbb{R}$ Borelian bounded):

$$P_t f(x) = \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi t}} \exp\left(-\frac{1}{2}\frac{(x-y)^2}{t}\right) f(y)dy \tag{12.27}$$

Or equivalently:

$$\forall A \in \mathcal{B}_{\mathbb{R}},\ P_t(x, A) = \int_A \frac{1}{\sqrt{2\pi t}} \exp\left(-\frac{1}{2}\frac{(x-y)^2}{t}\right) dy \tag{12.28}$$

That is, $P_t(x, dy)$ is the Gaussian measure centered in x, of variance t.

Proof for Theorem.

see [16] p93 ∎

13

Stochastic Calculus

The idea is to consider stochastic processes Y_t whose "infinitesimal increments" dY_t (for $t \in [a,b]$) are of the form $dY_t = X_t dB_t$, where dB_t is the infinitesimal increment of the Brownian motion B, and $X = (X_t)_{t\in[a,b]}$ is a process adapted to the filtration $(\mathcal{F}_t)_{t\in[a,b]}$ and "smooth enough". The infinitesimal Brownian increment dB_t has a non-null quadratic variation, which will lead eventually to the Itô's formula. The stochastic integral is defined on elementary stochastic processes, then extended to broader classes of stochastic processes.

Definition 13.0.1: Elementary process

A stochastic process $X = (X_s)_{s\in[a,b]}$ is called **elementary** if there exists a subdivision $a = t_0 < t_1 < ... < t_n = b$ of $[a,b]$, such that:

$$\forall t \in [a,b], \forall \omega \in \Omega, X_t(\omega) = \sum_{i=0}^{n-1} X_i(\omega)\mathbf{1}_{[t_i,t_{i+1}[}(t)$$

with $\forall i \in \{0,1,..,n-1\}, X_i$ is $\mathcal{F}_{t_i}$-measurable.
This means that, in each interval $[t_i, t_{i+1}[$, $X_t(\omega)$ is independent of t and $X_t(\omega) = X_i(\omega)$.
We define $\mathcal{E}$ (resp. $\mathcal{E}_n, n > 0$) the set of all elementary processes on $[a,b]$ (resp. the subset of the $X \in \mathcal{E}$)) such that all X_i have a finite moment $\mathbb{E}X_i < \infty$ (resp $\mathbb{E}(|X_i|^n) < \infty$

Definition 13.0.2: Stochastic integral of an elementary process

Let $X \in \mathcal{E}$, ie

$$X_t(\omega) = \sum_{i=0}^{n-1} X_i(\omega)\mathbf{1}_{[t_i,t_{i+1}[}(t)$$

The stochastic integral of X is the real random variable :

$$\int_a^b X_t dB_t := \sum_{i=0}^{n-1} X_i(B_{t_{i+1}} - B_{t_i})$$

Proposition 13.0.3

1. **linearity**
$$\forall X, Y \in \mathcal{E}, \forall \lambda, \mu \in \mathbb{R}, \int_a^b (\lambda X_t + \mu Y_t) dB_t = \lambda \int_a^b X_t dB_t + \mu \int_a^b Y_t dB_t$$
2. **centering**. If $X \in \mathcal{E}_1$, then $\int_a^b X_t dB_t \in L^1(\Omega, \mathcal{F}, \mathbb{P})$ and
$$\mathbb{E}\left(\int_a^b X_t dB_t\right) = 0$$
3. **membership in $L^2(\Omega, \mathcal{F}, \mathbb{P})$**. if $X \in \mathcal{E}_2$, then $\int_a^b X_t dB_t \in L^2(\Omega, \mathcal{F}, \mathbb{P})$, and
$$\mathbb{E}\left[\left(\int_a^b X_t dB_t\right)^2\right] = \mathbb{E}\left(\int_a^b X_t^2 dt\right)$$
4. **corollary**. The application I is an isometry:
$$I : \mathcal{E}_2 \subset L^2([a,b] \times \Omega, \mathcal{B}_{[a,b]} \otimes \mathcal{F}, dt \otimes d\mathbb{P}) \to L^2(\Omega, \mathcal{F}, \mathbb{P})$$
$$X \mapsto I(X) := \int_a^b X_t dB_t$$

Let $B = (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, (B_t)_{t\geq 0}, \mathbb{P})$ be a continuous Brownian motion.

Let $X = (X_t)_{t\in[a,b]}$ a stochastic process defined on $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, \mathbb{P})$, filtered space of B, restricted to $[a, b]$.

We define spaces of "smooth" stochastic processes X which allow to generalize the construction of the integral processes.

NB : in all the following, $\int_a^b X_t^2 dt$ is the random variable defined by $\forall \omega \in \Omega, \left(\int_a^b X_t^2 dt\right)(\omega) = \int_a^b X_t^2(\omega) dt$.

Definition 13.0.4: Space M^2

$X \in M^2$ if:

- X is progressively measurable [a]
- And:
$$\mathbb{E}\left(\int_a^b X_t^2 dt\right) < +\infty$$

M^2 is a Hilbert space, with norm:

$$\|X\|_{M^2} := \|X\|_{L^2([a,b]\times\Omega, dt\otimes\mathbb{P})} \tag{13.1}$$
$$\mathbb{E}\left(\int_a^b X_t^2 dt\right) = \int_{[a,b]\times\Omega} X_t(\omega)^2 dt d\mathbb{P} \tag{13.2}$$

[a] ie X is said to be progressively measurable w.r.t. a filtration $(\mathcal{F}_t)_{t\in T}$ if $\forall s \in T$, the application $(t, \omega) \mapsto X_t(\omega)$ is measurable from $([0, \mathbf{s}] \times \Omega, \mathcal{B}_{[0,s]} \times \mathcal{F}_s)$ to $(E, \mathcal{B}_E)$.

Definition 13.0.5: Space Λ^2

$X \in \Lambda^2$ if:

- X is progressively measurable
- and:

$$\int_a^b X_t^2 dt < \infty, \ \ \mathbb{P} - a.s.$$

Proposition 13.0.6

$\mathcal{E}_2 \subset M^2 \subset \Lambda^2$

Theorem 13.0.7: Density of $\mathcal{E}_2$ in M^2

$\mathcal{E}_2$, space of the square-integrable elementary processes, is a dense subspace of M^2 :

$$\forall X \in M^2, \exists (X^{(n)})_{n\in\mathbb{N}} \in \mathcal{E}_2, \text{s.t.} \lim_{n\to+\infty} \|X^{(n)} - X\|_{M^2} = 0$$

Proof for Theorem.

p122 ■

Theorem 13.0.8: Extension of the stochastic integral to M^2

For $X \in M^2$, we define $I(X) = \int_a^b X_t dB_t$ by:

$$\text{Let} (X^{(n)})_{n\in\mathbb{N}} \in \mathcal{E}_2, \ \text{s.t.} \lim_{n\to+\infty} \|X^{(n)} - X\|_{M^2} = 0$$

$$\lim_{n\to+\infty} \int_a^b X_t^{(n)} dB_t \underset{L^2}{=} \int_a^b X_t dB_t$$

Proof for Theorem.

p123 ■

This means that, to determine $\int X_t dB_t$ with $X \in M^2$, we need to find a sequence $X_t^{(n)} \in \mathcal{E}_2$ that converges (in L_2) towards X.

Proposition 13.0.9

Properties carry from $\mathcal{E}_n$ to M^2, most notably:

1. **centering**. If $X \in M^2$, then $\int_a^b X_t dB_t \in L^1(\Omega, \mathcal{F}, \mathbb{P})$ and
$$\mathbb{E}\left(\int_a^b X_t dB_t\right) = 0$$
2. **membership in $L^2(\Omega, \mathcal{F}, \mathbb{P})$**. if $X \in M^2$, then $\int_a^b X_t dB_t \in L^2(\Omega, \mathcal{F}, \mathbb{P})$, and
$$\mathbb{E}\left((\int_a^b X_t dB_t)^2\right) = \mathbb{E}\left(\int_a^b X_t^2 dt\right)$$
3. **covariance - dot product conservation by isometry** I. $\forall X, Y \in M^2$,
$$\mathbb{E}\left[\left(\int_a^b X_t dB_t\right)\left(\int_a^b Y_t dB_t\right)\right] = \mathbb{E}\left(\int_a^b X_t Y_t dt\right)$$

The extension of the stochastic integral to processes in Λ_2 also uses the convergence (this time, in probability) of a sequence $X_t^{(n)} \in \mathcal{E}$ towards X. Then the integrals $\int_a^b X_t^{(n)} dB_t$ converge towards $\int_a^b X_t dB_t$ also in probability. More formally:

Proposition 13.0.10

$\forall X \in \Lambda^2, \exists (X^{(n)})_{n\in\mathbb{N}} \in \mathcal{E}$ s.t.:
$$\lim_{n\to\infty} \int_a^b \left(X_t - X_t^{(n)}\right)^2 dt = 0, \ \mathbb{P} - a.s.$$
So $X_t^{(n)} \overset{\text{probability}}{\to} X_t$ also.

Theorem 13.0.11: Extension of the stochastic integral to Λ^2

For $X \in \Lambda^2$, with $(X^{(n)})_{n\in\mathbb{N}} \in \mathcal{E}$ st $\lim_{n\to\infty} \int_a^b \left(X_t - X_t^{(n)}\right)^2 dt = 0, \ \mathbb{P} - a.s.$, then the sequence of random variables $\int_a^b X_t^{(n)} dB_t$ converges in probability towards a random variable that is independent of $(X^{(n)})$:
$$\int_a^b X_t^{(n)} dB_t \underset{\text{probability}}{\longrightarrow} I(X) := \int_a^b X_t dB_t$$
And, if $X \in M^2$, this definition is coincident with the definition of the stochastic integral in M^2.

Proof for Theorem.

p126 ■

NB : recall that the convergence in probability is defined by:

$$Z_n \underset{\text{probability}}{\longrightarrow} Z \iff \forall \epsilon > 0, \lim_{n\to\infty} \mathbb{P}(|Z_n - Z| > \epsilon) = 0$$

Theorem 13.0.12: Sums of Riemann-Stieltjes

Let $X \in \Lambda^2$ be a **continuous** process. Then for all sequence of subdivisions $(\pi_n)_{n\in\mathbb{N}}$, $\pi_n = \{a = t_{n,0} < t_{n,1} < ... < t_{n,m_n} = b\}$, that verifies $|\pi_n| \underset{n\to\infty}{\to} 0$ (the "step' of the subdivisions converges to 0), then:

$$\sum_{i=0}^{m_n-1} X_{t_{n,i}}(B_{t_{n,i+1}} - B_{t_{n,i}}) \underset{n\to\infty}{\longrightarrow} \int_a^b X_t dB_t \quad \text{in probability}$$

Proof for Theorem.

p129 ∎

14

Ito's calculus and SDE

Assumption 14.0.1: context

In all the following,

- let $B = (\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, (B_t)_{t\geq 0}, \mathbb{P})$ be a continuous Brownian motion
- let $T > 0$ a fixed time
- $\Lambda^p(0,T), p \geq 1$ is the set of progressively measurable processes X that verify $\{t \mapsto X_t(\omega)\} \in L^p([0,T])$ $\mathbb{P}$-a.s.
- $M^2([0,T])$ the set of progressively measurable processes X such that $\mathbb{E}\left(\int_0^T X_t^2 dt\right) < \infty$ $\mathbb{P}$-a.s.
- we will always consider the continuous version of the stochastic integrals

Definition 14.0.2: Itô's process

A stochastic process $X = (X_t)_{t\geq 0}$ defined on $(\Omega, \mathcal{F}, \mathbb{P})$ and adpated to the filtration $(\mathcal{F}_t)_{t\in[0,T]}$ is called **Itô's process** if it exists two stochastic processes $a_s \in \Lambda^1([0,T])$ and $b_s \in \Lambda^2([0,T])$ such that:

$$\forall t \in [0,T], X_t = X_0 + \int_0^t a_s ds + \int_0^t b_s dB_s \tag{14.1}$$

Then we say that X **admits the stochastic differential** :

$$dX_t = a_t dt + b_t dB_t \tag{14.2}$$

The Itô's processes are stable by linear combination and multiplication - the Itô's processes set has an algebra structure.

Theorem 14.0.3: Stochastic differential of a product, integration by parts

Let X, Y be two Itô's processes on $[0, T]$:

$$dX_t = a_t^{(1)}dt + b_t^{(1)}dB_t$$
$$dY_t = a_t^{(2)}dt + b_t^{(2)}dB_t$$

Then $XY = (X_tY_t)_{t\in[0,T]}$ is also a Itô's process and:

$$d(X_tY_t) = X_tdY_t + Y_tdX_t + b_t^{(1)}b_t^{(2)}dt \tag{14.3}$$

The last term is the Itô's term.

Proof for Theorem.

book p155 for a clean proof, next chapter for a heuristic proof ■

Theorem 14.0.4: Itô's formula

An Itô's process remains an Itô's process when it is transformed by a deterministic function that is "smooth enough".
Let X be a Itô's process on $[0, T] : dX_t = a_tdt + b_tdB_t$.
Let:

$$f : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$$
$$(x, t) \mapsto f(x, t)$$

be $C^{2,1} : C^2$ in x, and C^1 in t.
Then $(f(X_t, t))_{t\in[0,T]}$ is also an Itô's process and:

$$d\left(f(X_t, t)\right) = \frac{\partial f}{\partial t}(X_t, t)dt + \frac{\partial f}{\partial x}(X_t, t)dX_t + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(X_t, t)b_t^2dt \tag{14.4}$$

The last term is Itô's complementary term.

Proof for Theorem.

see book p159 for a clean proof, and next chapter for a heuristic proof ■

Stochastic differential equations - SDE

Let $0 \le a < b \in \mathbb{R}$.

Definition 14.0.5: Stochastic differential equation

We call **stochastic differential equation (SDE)** on $[a, b]$, with the initial data ξ_a, a relation such as:

$$dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dB_t \tag{14.5}$$
$$X_a = \xi_a \tag{14.6}$$

where:

- X is an Itô's process on $[a, b]$ (the unknown)
- ξ_a is a given random variable, $\mathcal{F}_a$-measurable
- $\mu(x, t)$ and $\sigma(x, t)$ are two given functions defined over $\mathbb{R} \times [a, b] \to \mathbb{R}$

Solving the SDE is finding X Itô's process on $[a, b]$ such that:

$$X_t = \xi_a + \int_a^b \mu(X_s, s)ds + \int_a^b \sigma(X_s, s)dB_s \quad \forall t \in [a, b]$$

NB : a SDE may not have a solution.

Theorem 14.0.6: Existence and Unicity of a solution of a SDE

Let's assume

- $\sigma : t \mapsto \sigma(0, t)$ and $\mu : t \mapsto \mu(0, t)$ bounded on $[a, b]$
- σ and μ Lipschitz in x, y uniformly in t : $\exists c > 0$ st $\forall x, y \in \mathbb{R}, \forall t \in [a, b]$
  - $|\sigma(x, t) - \sigma(y, t)| \leq c|x - y|$
  - $|\mu(x, t) - \mu(y, t)| \leq c|x - y|$
- $\mathbb{E}(\xi_a^2) < \infty$

Then, there exists a solution $X = (X_t)_{t\in[a,b]} \in M^2$ of the SDE, and it unique (up to un-discernibility).

Proof for Theorem.

[16] p163 ■

Theorem 14.0.7: Stochastic exponential

$$dX_t = X_t dB_t$$
$$X_0 = 1$$

has for unique solution $X_t = e^{B_t - \frac{t}{2}}$ (see next chapter).
More generally, let Z_t be a Itô's process:

$$dZ_t = a_t dt + b_t dB_t$$
$$Z_0 = 0$$

Then the stochastic differential equation

$$dX_t = X_t dZ_t = a_t X_t dt + b_t X_t dB_t$$
$$X_0 = 1$$

has a unique solution called **the stochastic exponential of** Z:

$$\mathcal{E}(Z)(t) = \exp\left(Z_t - \frac{1}{2}\int_0^t b_s^2 ds\right) \quad \forall t \in [0, T]$$

Proof for Theorem.

[16] p184 ∎

Theorem 14.0.8: Numerical approximation of the solution of a SDE - Euler-Maruyama

Let X be a Itô's process solution of:

$$dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dB_t$$
$$X_a = \xi_a$$

where $\mu(X_t, t)$ is the **drift** and $\sigma^2(X_t, t)$ the **diffusion coefficient**.
Then one can approximate numerically the solution of the SDE by:

$$a = t_0 < t_1 < ... < t_k < t_{k+1} < ... < t_n = b$$
$$x^{(k+1)} = x^{(k)} + \mu(x^{(k)}, t_k)\Delta t + \sigma(x^{(k)}, t_k)\sqrt{\Delta t}\,\mathcal{N}(0, 1)$$
$$t_{k+1} = t_k + \Delta t$$

Proof for Theorem.

We have $\Delta X_t \sim \mu(X_t, t)\Delta t + \sigma(X_t, t)\Delta B_t$, where $\Delta B_t \sim \mathcal{N}(0, \Delta t)$. So $\sigma(X_t, t)\Delta B_t \sim \mathcal{N}(0, \sigma^2(X_t, t)\Delta t)$, and $\Delta X_t \sim \mathcal{N}(\mu(X_t, t), \sigma^2(X_t, t)\Delta t)$ ∎

SDE solutions are Markov Processes

Intuitively, when considering the solution X_t of 14.5 between t and $t + \Delta_t$, leads to:

$$X_{t+\Delta t} = X_t + \sigma(X_t, t)\Delta B_t + \mu(X_t, t)dt$$

We know that ΔB_t is independent of $\mathcal{F}_t$, so $X_{t+\Delta t}$ depends on the past only by X_t. This suggest $(X_t)t \geq 0$ is a Markov

Process.

We actually have the following theorem:

Theorem 14.0.9: SDE solutions are Markov Processes

Consider the following SDE:

$$\begin{aligned} dX_t &= \mu(X_t, t)dt + \sigma(X_t, t)dB_t \\ X_0 &= \xi_0 \end{aligned}$$

where σ and μ verify the hypothesis in 14. The solution X_t is given by:

$$X_t = \xi_0 + \int_0^t \mu(X_u, u)du + \int_0^t \sigma(X_u, u)dB_u$$

Then **X is Markov process**, with transition kernels given by ($\forall x \in \mathbb{R}, \forall A \in \mathcal{B}_{\mathbb{R}}, \forall t \geq s$):

$$\begin{aligned} P_{s,t}(x, A) &= \mathbb{P}(X_t^{x,s} \in A) \\ X_t^{x,s} &= x + \int_s^t \mu(X_u^{x,s}, u)du + \int_s^t \sigma(X_u^{x,s}, u)dB_u \end{aligned}$$

ie $X^{x,s} = (X_t^{x,s})_{t \geq s}$ is the solution of the SDE starting from x at time t.
If the SDE is time-invariant, ie

$$dX_t = \mu(X_t)dt + \sigma(X_t)dB_t \tag{14.7}$$
$$X_0 = \xi_0 \tag{14.8}$$

then the Markov process solution X is homogeneous. (ie $P_{s,t}$ depends only on $t - s$). ie (for f Borelian bounded):

$$P_t f(x) = \int_{\mathbb{R}} f(z)P_t(x, dz) = \int_{\mathbb{R}} f(z)\mathbb{P}(X_t^{x,0} \in dz)$$

Proof for Theorem.

[16] p173 ∎

We can compute the infinitesimal generator when f is smooth enough:

Theorem 14.0.10: Infinitesimal generator of X

In the case of time-invariant SDE 14.7 (when X is homogeneous), then : for any $f \in C^2(\mathbb{R})$ bounded, with derivatives bounded, for any $t > 0$, for any $x \in \mathbb{R}$:

$$P_t f(x) = f(x) + \int_0^t P_s(Af)(x)ds \tag{14.9}$$
$$Af(x) = \frac{1}{2}\sigma^2(x)f''(x) + \mu(x)f'(x) \tag{14.10}$$

Proof for Theorem.

∎

When the probability transitions have densities, we end up with the Kolmogorov equations:

Theorem 14.0.11: Kolmogorov equations for a general SDE

Consider SDE:

$$dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dB_t$$
$$X_0 = \xi_0$$

We assume the transition probabilities have densities:

$$P_t(x, dy) = p_t(x, y)dy \quad (x, y) \in \mathbb{R}$$

Then, remembering that x is the "start" and y the "arrival":

- $\frac{\partial}{\partial t}p_t(x, y) = \left(\frac{1}{2}\sigma(x)^2\frac{\partial^2}{\partial x^2} + \mu(x)\frac{\partial}{\partial x}\right)p_t(x, y)$
- $\frac{\partial}{\partial t}p_t(x, y) = \frac{1}{2}\frac{\partial^2}{\partial y^2}(\sigma(y)^2 p_t(x, y)) - \frac{\partial}{\partial y}(\mu(y)p_t(x, y))$

The second equation ("futur Kolmogorov") is also known as the Fokker-Plank equation.

Proof for Theorem.

15

Quantifying randomness of data sequences

In its most general form, a data sequence can be considered as a realization of a stochastic process. A stochastic process is usually described as either a sequence of random variables (ie a set of $(X_t)_{t\geq 0} : \omega \to X_t(\omega)$), or a single random variable over the space of functions (ie $X : \omega \to X(\omega) = X_\omega = \{t \to X_\omega(t)\}$.

As we wish to measure -somehow- the degree of "randomness" of a data sequence, we will find more convenient to use the former view (ie consider a data sequence as a countable sequence of random variables), as it allows to use the framework of information theory.

First, we will recall the basic definitions of information theory : entropy, relative entropy (ie KL divergence), conditional entropy and mutual information. Then we will write some results regarding the application of Information Theory (IT) to data sequence. Last, we will describe two of the most used empirical measurements : Approximate Entropy (ApEn) and Sample Entropy (SampEn).

The basic of IT are introduced in [**bishop˙pattern˙2016**], [13], and [17]. One of the reference books on the subject is [6], and goes much further than the scope of this report. Of course, the interested reader will also refer to the seminal paper by Shannon : [26].

Entropy : given a random variable X, either discrete or continuous, taking values in a measurable space $\mathcal{X}$, and its probability distribution p, the amount of information given by a given realization x is given by $\log \frac{1}{p(x)} = -\log p(x)$ (the lower the probability, the higher the amount of information).

The average amount of information (over all possible values of x) required to describe the random variable X is the entropy of X :

$$\mathcal{H}(X) = -\sum_{x\in\mathcal{X}} p(x) \log p(x) \tag{15.1}$$

(or $-\int_{\mathcal{X}} p(x) \log p(x) dx$).

Relative entropy, KL divergence : when approximating the true data distribution p by a distribution q, we require in average a quantity of information $-\sum_{x\in\mathcal{X}} p(x) \log q(x)$ to describe X. The difference between the optimal amount of information (ie the entropy $\mathcal{H}(X)$) and this quantity is the well-known relative entropy, of KL-divergence between

p and q :

$$\mathbb{KL}(p\|q) = \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)} \tag{15.2}$$

$$= \int_{\mathcal{X}} p(x)\log\frac{p(x)}{q(x)}dx \tag{15.3}$$

The properties of KL-divergence (positiveness, non symmetry) are well described in the sources above.

Conditional Entropy : we now consider two random variables X and Y, and wish to measure the degree of relationship between them. We define the conditional entropy of, for example, Y given X, as the amount of information we get observing the values of Y given X, averaged over the joint probability. Formally :

$$\mathcal{H}(Y|X) = -\sum_{x,y\in\mathcal{X},\mathcal{Y}} p(x,y)\log p(y|x)$$

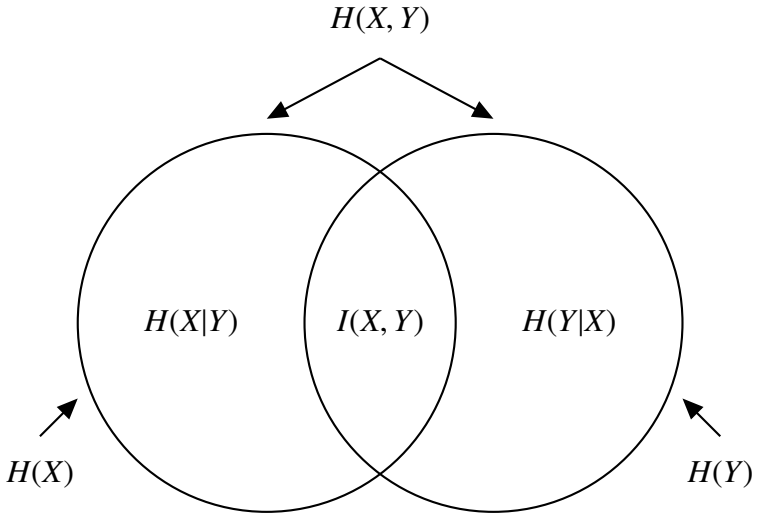
$$= -\int_{\mathcal{X},\mathcal{Y}} p(x,y)\log p(y|x)\,dxdy$$

By basic calculation, we get $\mathcal{H}(X,Y) = \mathcal{H}(Y|X) + \mathcal{H}(X) = \mathcal{H}(X|Y) + \mathcal{H}(Y)$

Mutual Information - last, still considering two random variables X, Y, the mutual information is the additional amount of information we need to describe X, Y when we assume independence (ie use $p(x)p(y)$) rather than use the true joint probability $p(x,y)$. This amount is $-\log p(x)p(y) - (-\log p(x,y)) = \log\frac{p(x,y)}{p(x)p(y)}$, that we average over the true distribution $p(x,y)$:

$$\mathcal{I}(X,Y) = \sum_{x,y\in\mathcal{X},\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \tag{15.4}$$

The relationships between entropy, relative entropy, conditional entropy and mutual information, are well described using a Venn diagram:



For example : $\mathcal{I}(X,Y) = \mathcal{H}(X) + \mathcal{H}(Y) - \mathcal{H}(X,Y)$, etc.

If we now consider a sequence of n random variables, a way to describe the randomness of the sequence is to measure how the entropy of the joint distribution grow with n. (see [6] p63). Typically, if the random variables are independent, we can expect the entropy to grow at each step n by the full amount of $\mathcal{H}(X_n)$. On the other hand, if

correlations exist, then we can expect the overall entropy to grow by a lesser amount.

Formally, the **entropy rate** of a stochastic process $(X_n)_{n\in\mathbb{N}^*}$ is defined by:

$$\mathcal{H}(X) = \lim_{n\to\infty} \frac{1}{n}\mathcal{H}(X_1, X_2, ..., X_n) \tag{15.5}$$

when the limit exists. This is the entropy per symbol.

[6] also defines a **conditional entropy rate**

$$\mathcal{H}'(X) = \lim_{n\to\infty} \frac{1}{n}\mathcal{H}(X_n|X_{n-1}, ...X_2, X_1) \tag{15.6}$$

when the limit exists. This is the entropy of the latest random variable, given the past realizations.

An important result is that, when (X_n) is stationary, both limits exist and are equal.

The question naturally arises that, given a data sequence, how do we compute an approximation of 15.5 or 15.6. We now introduce the ApEn and SampEn (see [7] and [20]).

Approximate Entropy - Approximate Entropy is a measure of the log probability that patterns, that were identified in the data through the examination of sub sequences of a given length, still remain when considering longer subsequences.

Formally, let $x = (x_1, x_2, ..., x_N)$ be a data sequence of length N, m an integer $(0 < m \leq N)$, and $r > 0$ a measure of acceptable noise. We define as *blocks* of length m the subsequences $b_i^m = \{x_i, x_{i+1}, ..., x_{i+m-1}\}$, starting at i with $(1 \leq i \leq N - m + 1)$. For two blocks b_i^m and b_j^m, we define a component-wise distance $d_{ij}^m = \max_{k=0,1,..,m-1} |b_{i+k}^m - b_{j+k}^m|$. We consider "close enough" (ie similar) those blocks whose distance is lower than the acceptable noise (tolerance) r, and calculate the frequency of those similar blocks (b_i^m, b_j^m) w.r.t. all blocks of length m for a given b_i^m :

$$C_i^m(r) = \frac{\text{number of } j \leq N - m + 1 \text{ s.t. } d_{ij} \leq r}{\text{number of blocks of length } m : N - m + 1} \tag{15.7}$$

We then average the $\log C_i^m$ over all possible subsequences b_i :

$$\Phi^m(r) = \frac{1}{N - m + 1} \sum_{i=1}^{N-m+1} \log C_i^m(r)$$

We can see $\Phi^m(r)$ as the average of the log probability of two subsequences of length m to be similar (up to the tolerance r). Finally, we compute:

$$\text{ApEn}(m, r, N) = \Phi^m(r) - \Phi^{m+1}(r) \tag{15.8}$$

When $m \ll N$, then $-\text{ApEn}(m, r, N) \sim \frac{1}{N+1} \sum_{i=1}^{N+1} \log \frac{C_i^{m+1}}{C_i^m}$, which is the average over i of the log of the conditional probability of two sequences b_i^{m+1}, b_j^{m+1} of length $m + 1$ be similar given that they are already similar for lengths $1, 2, ..., m$.

Last, $\text{ApEn}(m, r, N)$ is a statistical estimator of:

$$\text{ApEn}(m, r) = \lim_{N\to\infty} \text{ApEn}(m, r, N) \tag{15.9}$$

Intuitively, the greater the regularity in a data sequence, the greater the likelihood that patterns existing for subsequences of length m still remain for subsequences of greater length, ie the smaller ApEn, and conversely.

Key properties of ApEn include:

- ApEn is independent of any model of the data sequence.
- due to its construction, ApEn is non-negative, is finite for stochastic processes and deterministic processes with noise.
- following [20], it is imperative to eliminate any trend in the data sequence before computing ApEn and drawing conclusions.
- typical recommended values for m are 2 and 3. Typical recommended values for r are in the range of 0.1 to 0.25 the standard deviation of the data sequence, in order to allow a sufficient number of subsequences close within a distance r, and reasonable estimates of the conditional probabilities.
- if the noise is significant (ie signal-to-noise ratio lower than three), then precautions must be taken with the interpretation of ApEn.
- Regular measurements of the data sequence over time are required.
- Normalization of data sequences is required before computations of ApEn to compare data sequences between each other.

Sample Entropy - we can see in 15.7 that a given subsequence b_i^m is counted in both the numerator and the denominator. If this ensures numerical stability (ie no attempt to calculate $\log 0$ for example), this also introduces a bias in the calculation of the probability estimates. SampEn explicitly discounts the subsequence b_i^m from the calculations to remove the bias.

Formally:

$$A_i^m(r) = \frac{1}{N-m-1}\{\text{number of blocks } b_j^{m+1} \text{ of length } m+1 \text{ s.t. } i \neq j \text{ and } d_{ij}^{m+1} \leq r\} \tag{15.10}$$

$$B_i^m(r) = \frac{1}{N-m}\{\text{number of blocks } b_j^{m} \text{ of length } m \text{ s.t. } i \neq j \text{ and } d_{ij}^{m} \leq r\} \tag{15.11}$$

$$B^m(r) = \frac{1}{N-m}\sum_{i=1}^{N-m} B_i^m(r) \tag{15.12}$$

$$A^m(r) = \frac{1}{N-m-1}\sum_{i=1}^{N-m-1} A_i^m(r) \tag{15.13}$$

And

$$\text{SampEn}(m, r, N) = -\log \frac{A^m(r)}{B^m(r)} \tag{15.14}$$

$$\text{SampEn}(m, r) = -\lim_{n\to\infty} \log \frac{A^m(r)}{B^m(r)} \tag{15.15}$$

16

Neural ODE and SDE

Neural-ODEs is a class of models introduced in [3] : "Neural Ordinary Differential Equations" by Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, David Duvenaud. ArXiV : Neural ODE Best Paper Award NeurIPS 2018.

The starting point is to write the evolution of the latent state z_t as:

$$z_{t+1} = z_t + f(z_t, \theta_t) \tag{16.1}$$

where $z_t \in \mathbb{R}^D$ is the latent state, θ_t is a set of parameters at time t, and f is a function.

This formulation is the one used in ResNet blocks, and can be seen as the Euler transformation of a continuous transformation.

Taking the expression to the limit as $dt \to 0$, we can write an ODE:

$$\frac{dz_t}{dt} = f(z_t, t, \theta_f) \tag{16.2}$$

where θ_f is a set of parameters, that can typically be the parameters of a neural network learning f.

For a time series $x_{t_1}, x_{t_2}, ...x_{t_N}$, Chen and al. in [3] assume the following generative model:

$$z_{t_0} \sim p_{\theta_z}(z_{t_0}) \tag{16.3}$$
$$z_{t_1}, z_{t_2}, ..., z_{t_N} = \text{ODE Solver}(z_{t_0}, f, \theta_f, t_0, ..., t_N) \tag{16.4}$$
$$x_{t_i} \sim p_{\theta_x}(x_t|z_t) \tag{16.5}$$

We note that the latent variable is stochastic only through its initial state z_{t_0}. The evolution of z_t is then deterministic through the ODE.

The inference model is:

$$[\mu_\phi, \Sigma_\phi] = \text{LSTM}(x_{t_0:t_N}) \tag{16.6}$$
$$q_\phi(z_{t_0}|x_{t_0:t_N}) = \mathcal{N}(z_{t_0}|\mu_\phi, \Sigma_\phi) \tag{16.7}$$

We reproduce here the drawing from the paper:

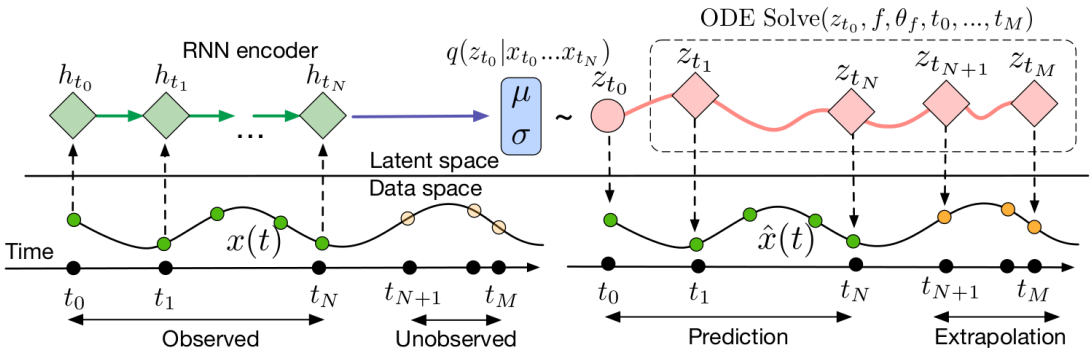


Figure 16.1: Neural ODE model

The model is trained maximizing a VLB as usual.

A key point is then to be able to compute the gradients of the network f with respect to θ_f. One method is the **adjoint sensitivity method** described in [21], and that the interested reader will find in the appendix F

A limitation of this model is the assumption that the prior is "concentrated" in the initial value z_{t_0}, and that the remaining latent variables are deterministically determined.

One can then modify the latent model from an ODE to a SDE:

$$dZ_t = f_{\theta_f}(Z_t, t)dt + L(Z_t, t)dB_t \tag{16.8}$$

with the notations that we know well now. The adjoint sensitivity method carries over the SDE, even though I am not fully knowledgeable on this yet.

Appendices

A

Vanilla Variational Auto Encoder

We consider a sequence of i.i.d points $(x_i)_{i=1,\dots,N} \in \mathbb{R}^D$, and the associated latent variables $(z_i)_{i=1,\dots,N} \in \mathbb{R}^L$.

In the vanilla VAE setting, the observation model (decoder) is $p_{\theta_x}(x|z)$, the approximate posterior (encoder) is $q_\phi(z|x)$, the latent prior is $p_{\theta_z}(z)$.

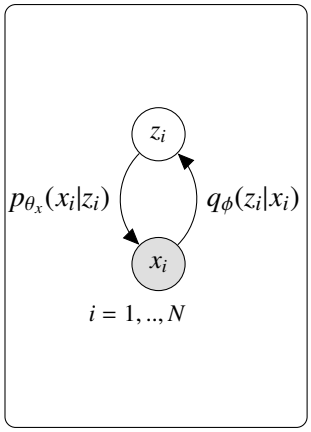


Figure A.1: Vanilla VAE

The log likelihood of the data is:

$$\log p_\theta(x) = \log \frac{p(x,z)}{p(z|x)}$$

Multiplying both sides by $q_\phi(z|x)$ and integrating over dz leads to:

$$\begin{aligned}
\log p_\theta(x) &= \int q_\phi(z|x) \log \frac{p_\theta(x,z)}{p(z|x)} dz \\
&= \int q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} \frac{q_\phi(z|x)}{p(z|x)} dz \\
&= \mathbb{E}_{q_\phi(z|x)} \log \frac{p_\theta(x,z)}{q_\phi(z|x)} + \mathbb{KL}(q_\phi(z|x)||p(z|x)) \\
&\geq \mathbb{E}_{q_\phi(z|x)} \log \frac{p_\theta(x,z)}{q_\phi(z|x)} = \mathcal{L}(\theta, \phi, X)
\end{aligned}$$

In this setting, the D-separation is obvious and the joint distribution factorizes over n:

$$p_\theta(x, z) = \prod_{i=1}^{n} p_{\theta_x}(x_i|z_i)p_{\theta_z}(z_i)$$
$$q_\phi(z|x) = \prod_{i=1}^{n} q_\phi(z_i|x_i)$$

The VLB (or ELBO) $\mathcal{L}(\theta, \phi, X)$ simplifies into:

$$\begin{aligned}\mathcal{L}(\theta, \phi, X) &= \mathbb{E}_{q_\phi(z|x)} \log \frac{\prod_{i=1}^{n} p_{\theta_x}(x_i|z_i)p_{\theta_z}(z_i)}{\prod_{i=1}^{n} q_\phi(z_i|x_i)} \\ &= \sum_{i=1}^{n} \mathbb{E}_{q_\phi(z_i|x_i)} \log p_{\theta_x}(x_i|z_i) - \sum_{i=1}^{n} \mathbb{KL}(q_\phi(z_i|x_i)||p_{\theta_z}(z_i))\end{aligned}$$

The first term is the reconstruction loss, and is estimated via Monte Carlo sampling over $z_i \sim q_\phi(z_i|x_i)$. The second term is a KL-divergence, which can be computed analytically when q_ϕ and p_{θ_z} are chosen to be Gaussians.

B

Gaussian Process

We summarize here most of the results of the Gaussian Process, and refers the reader to [22] for further details.

We first recall the Gaussian marginal and conditional result:

Let x and y be jointly Gaussian vectors, ie:

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} A & C \\ C^T & B \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \tilde{A} & \tilde{C} \\ \tilde{C}^T & \tilde{B} \end{bmatrix}^{-1}\right) \tag{B.1}$$

where A, B, C is the block decomposition of the covariance matrix, and $\tilde{A}, \tilde{B}, \tilde{C}$ the block decomposition of the precision matrix.

Then the marginal distribution of x and the conditional distribution of x given y are :

$$x \sim \mathcal{N}(\mu_x, A) \tag{B.2}$$

$$x|y \sim \mathcal{N}(\mu_x + CB^{-1}(y - \mu_y), A - CB^{-1}C^T) \tag{B.3}$$

$$= \mathcal{N}(\mu_x - \tilde{A}^{-1}\tilde{C}(y - \mu_y), \tilde{A}^{-1}) \tag{B.4}$$

We now consider a Gaussian Process with mean function $m(.)$ and kernel $k(.,.)$

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{B.5}$$

At the training points $X = \{x_1, ..., x_n\}$, the observations are $Y = \{y_1, ..., y_n\}$ with some noise $y = f(x) + \epsilon$ with $\epsilon \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma_n^2)$.

The covariance between observations writes:

$$\text{cov}(y_p, y_q) = k(x_p, x_q) + \delta_{pd}\sigma_n^2 \tag{B.6}$$

$$\text{cov}(y) = K(X, X) + \sigma_n^2 I \tag{B.7}$$

At some test points X_*, we aim to predict $f_* = f(X_*)$. Then:

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right) \tag{B.8}$$

From which we get:

$$f_*|X_*, X, Y \sim \mathcal{N}(\overline{f_*}, \mathrm{cov}(\mathrm{f}_*)) \tag{B.9}$$

$$\overline{f_*} = K(X_*, X)\left(K(X, X) + \sigma_n^2 I\right)^{-1} Y \tag{B.10}$$

$$\mathrm{cov}(\mathrm{f}_*) = K(X_*, X_*) - K(X_*, X)\left(K(X, X) + \sigma_n^2 I\right)^{-1} K(X, X_*) \tag{B.11}$$

C

KL divergence between two exponential-family distributions

We recall the family of distributions parameterized by $\eta \in \mathbb{R}^K$, over a fixed support $\mathcal{X}^D \in \mathbb{R}^D$: the **exponential family** of distributions $p(x|\eta)$ is given by:

$$p(x|\eta) = \frac{1}{Z(\eta)} h(x) \exp\left(\eta^T \mathcal{T}(x)\right) \tag{C.1}$$

$$= h(x) \exp\left(\eta^T \mathcal{T}(x) - A(\eta)\right) \tag{C.2}$$

with:

- $h(x)$ is the base measure, ie a scaling constant (often 1)
- $\mathcal{T}(x)$ are the sufficient statistics
- η are the natural parameters, or canonical parameters
- $Z(\eta)$ is the partition function, $A(\eta)$ is the log partition function.

The Bernoulli, categorical (ie multinomial for one observation), Gaussian distributions are part of the exponential family.

The $\mathbb{KL}$**-divergence between two exponential family distributions of the same family** is:

$$\mathbb{KL}(p(x|\eta_1)||p(x|\eta_2)) = \mathbb{E}_{\eta_1}\left[(\eta_1 - \eta_2)\mathcal{T}(x) - A(\eta_1) + A(\eta_2)\right] \tag{C.3}$$

$$= (\eta_1 - \eta_2)^T \mathbb{E}_{\eta_1} \mathcal{T}(x) - A(\eta_1) + A(\eta_2) \tag{C.4}$$

The most important example is the $\mathbb{KL}$-divergence between two multivariate Gaussian distributions of dimension D:

KL between two multivariate Gaussians of dimension D

$$\mathbb{KL}(\mathcal{N}(x|\mu_1, \Sigma_1)||\mathcal{N}(x|\mu_2, \Sigma_2) = \frac{1}{2}\left[\text{tr}(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^T \Sigma_2^{-1}(\mu_2 - \mu_1) - D + \log \frac{|\Sigma_2|}{|\Sigma_1|}\right] \tag{C.5}$$

D

Ornstein Uhlenbeck

We summarize here some computations of chapters 8 and 9 for the Ornstein-Uhlenbeck process.

We start by a preliminary result:

Theorem D.0.1: Stochastic exponential

Let Z_t be the Itô's process (ie a_t and b_t two stochastic processes)

$$\begin{aligned} dZ_t &= a_t dt + b_t dB_t \\ Z_0 &= 0 \end{aligned}$$

Then consider the SDE :

$$\begin{aligned} dX_t &= X_t dZ_t \\ &= a_t X_t dt + b_t X_t dB_t \\ X_0 &= 1 \end{aligned}$$

Then the solution is:

$$X_t = e^{Y_t} \tag{D.1}$$

$$Y_t = Z_t - \frac{1}{2}\int_0^t b_s^2 ds \tag{D.2}$$

Proof for Theorem.

We set

$$\begin{aligned} X_t &= e^{Y_t} = f(Y_t, t) \\ dY_t &= \mu_t dt + \sigma_t dB_t \end{aligned}$$

and apply Itô's formula:

$$\begin{aligned} dX_t &= d(e^{Y_t}) \\ &= X_t dY_t + \frac{1}{2} X_t \sigma_t^2 dt \\ &= X_t \left(\mu_t dt + \sigma_t dB_t\right) + \frac{1}{2} X_t \sigma_t^2 dt \\ &= \left(X_t \mu_t + \frac{1}{2} X_t \sigma_t^2\right) dt + X_t \sigma_t dB_t \end{aligned}$$

By identification:

$$\begin{aligned} \mu_t &= a_t - \frac{1}{2}\sigma_t^2 \\ \sigma_t &= b_t \end{aligned}$$

Then

$$\begin{aligned} dY_t &= \left(a_t - \frac{1}{2}\sigma_t^2\right) dt + \sigma_t dB_t \\ &= dZ_t - \frac{1}{2}\sigma_t^2 dt \\ Y_t &= Z_t - \frac{1}{2}\int_0^t b_s^2 ds \ (Y_0 = 0) \end{aligned}$$

Conversely, e^{Y_t} is solution, as

$$\begin{aligned} d(e^{Y_t}) &= e^{Y_t} dY_t + \frac{1}{2} e^{Y_t} b_t^2 dt \\ &= e^{Y_t}\left(dY_t + \frac{1}{2} b_t^2 dt\right) \\ &= e^{Y_t} dZ_t \end{aligned}$$

■

Definition D.0.2: Ornstein Uhlenbeck process

The **Ornstein Uhlenbeck process** is the stochastic processes solution of the linear SDE:

$$dX_t = -\lambda X_t dt + \sigma dB_t \tag{D.3}$$

$$X_{t_0} = X_0 \tag{D.4}$$

where $\lambda > 0$ is the **drift** and $\sigma > 0$ the **diffusion coefficient**.

Theorem D.0.3: Solution of Ornstein Uhlenbeck

$$dX_t = -\lambda X_t dt + \sigma dB_t \tag{D.5}$$
$$X_{t_0} = X_0 \tag{D.6}$$
$$X_t = e^{-\lambda t} X_0 + \sigma \sqrt{\frac{1 - e^{-2\lambda t}}{2\lambda}} \mathcal{N}(0, 1) \tag{D.7}$$

Proof for Theorem.

For Ornstein-Uhlenbeck, we write $Y_t = e^{\lambda t} X_t$. Ito's formula gives :

$$dY_t = e^{\lambda t} X_t dX_t + \lambda e^{\lambda t} X_t \tag{D.8}$$
$$= \sigma e^{\lambda t} dB_t \tag{D.9}$$
$$Y_t = Y_0 + \sigma \int_0^t e^{\lambda s} dB_s \tag{D.10}$$
$$X_t = e^{-\lambda t} X_0 + \sigma e^{-\lambda t} \int_0^t e^{\lambda s} dB_s \tag{D.11}$$

We have to compute:

$$\int_0^t e^{\lambda s} dB_s = \lim_{n \to \infty} \sum_{k=0}^{n} e^{\lambda t_k} (B_{t_{k+1}} - B_{t_k}) \tag{D.12}$$

The sum on the rhs is a sum of independent Gaussians $e^{\lambda t_k}(B_{t_{k+1}} - B_{t_k}) \sim e^{\lambda t_k} \mathcal{N}(0, t_{k+1} - t_k)$, so the sum is a centered Gaussian of variance $\sum_{k=0}^{n} e^{2\lambda t_k}(t_{k+1} - t_k)$. So :

$$\int_0^t e^{\lambda s} dB_s = \mathcal{N}(0, \lim_{n \to \infty} \sum_{k=0}^{n} e^{2\lambda t_k}(t_{k+1} - t_k)) \tag{D.13}$$
$$= \mathcal{N}(0, \int_0^t e^{2\lambda s} ds) \tag{D.14}$$
$$= \mathcal{N}(0, \frac{1}{2\lambda}(e^{2\lambda t} - 1)) \tag{D.15}$$

Theorem D.0.4: Fokker Plank Kolmogorov solution for O.U.

The stationary solution of the Fokker Plank Kolmogorov equation of a 1D Ornstein Uhlenbeck process is:

$$\frac{\partial^2 p}{\partial x^2} + \frac{2\lambda}{\sigma} x \frac{\partial p}{\partial x} + \frac{2\lambda}{\sigma} p = 0 \tag{D.16}$$
$$p \propto \exp - \frac{\lambda x^2}{\sigma} \tag{D.17}$$

Proof for Theorem.

We have $F(X_t, t) = -\lambda X_t$, $L(X_t, t) = 1$, $Q = \sigma$ in 9.4, which leads to:

$$\frac{\partial p}{\partial t} = \frac{\sigma}{2}\frac{\partial^2 p}{\partial x^2} + \lambda x \frac{\partial p}{\partial x} + \lambda p \tag{D.18}$$

When p is stationary (ie when $t \to \infty$), then $\frac{\partial p}{\partial t} = 0$ and we get the result. ∎

Theorem D.0.5: Moment differential equations for O.U.

The equations 9.20 simplify in:

$$\frac{dm}{dt} = -\lambda m \tag{D.19}$$
$$\frac{dP}{dt} = -2\lambda P + \sigma \tag{D.20}$$
$$x(0) = x_0 \tag{D.21}$$
$$P(0) = 0 \tag{D.22}$$
$$X_t \sim \mathcal{N}(X_t | m(t), P(t)) \tag{D.23}$$

Proof for Theorem.

substituting λ and σ in 9.20 ∎

Theorem D.0.6: Discretization of O.U

The discretization 9.26 writes:

$$x_{t_{k+1}} = a_k x_{t_k} + q_k \tag{D.24}$$
$$q_k \sim \mathcal{N}(0, \Sigma_k) \tag{D.25}$$
$$a_k = e^{-\lambda \Delta t_k} \tag{D.26}$$
$$\Sigma_k = \frac{\sigma}{2\lambda}\left(1 - e^{-2\lambda \Delta t_k}\right) \tag{D.27}$$

Proof for Theorem.

One can substitute λ and σ into 9.26.

For $\lambda > 0$ and $\sigma > 0$, we can write from D.9 (with $t_{k+1} = t_k + \Delta t$):

$$Y_{t_{k+1}} - Y_{t_k} = \sigma \int_{t_k}^{t_{k+1}} e^{\lambda s} dB_s \tag{D.28}$$
$$= \sigma e^{\lambda t_k} \mathcal{N}(0, \frac{e^{2\lambda \Delta t} - 1}{2\lambda}) \tag{D.29}$$
$$X_{t_{k+1}} = e^{-\lambda t_{k+1}} Y_{t_{k+1}} \tag{D.30}$$
$$= e^{-\lambda \Delta t} X_{t_k} + \sigma \sqrt{\frac{1 - e^{-2\lambda \Delta t}}{2\lambda}} \mathcal{N}(0, 1) \tag{D.31}$$

In other words:

$$p(X_{t_{k+1}} | X_{t_k}) = \mathcal{N}\left(X_{t_{k+1}} | e^{-\lambda \Delta t} X_{t_k}, \sigma^2 \frac{1 - e^{-2\lambda \Delta t}}{2\lambda}\right) \tag{D.32}$$

Theorem D.0.7: Maximum Likelihood parameters estimation

Let $x_0, x_1, ...x_k, ...x_n$ be an observation of a Ornstein Uhlenbeck process with unknown $\lambda > 0$ and $\sigma > 0$. The previous results lead to the following maximum likelihood point estimates:

$$\lambda_{ML} = -\frac{1}{\Delta t} \log \frac{\sum_{k=0}^{n-1} x_k x_{k+1}}{\sum_{k=0}^{n-1} x_k^2} \tag{D.33}$$

$$\sigma_{ML}^2 = \frac{1}{n} \frac{2\lambda_{ML}}{1 - e^{-2\lambda_{ML}\Delta t}} \sum_{k=0}^{n-1} \left(x_{k+1} - e^{-\lambda_{ML}\Delta t} x_k \right)^2 \tag{D.34}$$

Proof for Theorem.

We have :

$$p(x_{k+1}|x_k) = \mathcal{N}\left(x_{k+1} | e^{-\lambda \Delta t} x_k, \sigma^2 \frac{1 - e^{-2\lambda \Delta t}}{2\lambda} \right) \tag{D.35}$$

With:

$$a = e^{-\lambda \Delta t} \tag{D.36}$$

$$\eta^2 = \sigma^2 \frac{1 - e^{-2\lambda \Delta t}}{2\lambda} \tag{D.37}$$

The likelihood writes:

$$p(x_{1:n}|\lambda, \sigma) = \prod_{k=0}^{n-1} p(x_{k+1}|x_k) \tag{D.38}$$

$$-\log p(x_{1:n}|\lambda, \sigma) = \frac{n}{2} \log 2\pi\eta^2 + \frac{1}{2\eta^2} \sum_{k=0}^{n-1} (x_{k+1} - a x_k)^2 \tag{D.39}$$

Forming the gradient and putting it to zero gives the result. ∎

E

Why Brownian motion is a Gaussian and a Markov process

Solutions of linear SDE are Gaussian processes and Markov processes. I found counter-intuitive at first, that a Gaussian process might be Markovian. After all, the kernel function encodes dependencies and correlations between several points, so there is no *a priori* reason to have $p(x_n|x_{n-1}, x_{n-2}, ..., x_1) = p(x_n|x_{n-1})$.

The Markovian property is actually enabled by the GP kernel. Here is a toy example for the Browian motion.

The Brownian motion is solution of the simplest linear equation: $dX_t = dB_t$.

Theorem E.0.1: Kernel function of the Brownian motion

The Brownian motion is the solution of the linear SDE

$$dX_t = dB_t \tag{E.1}$$

It is a Gaussian process with mean and kernel functions given by:

$$B_t \sim \mathcal{GP}(0, \min(t, t')) \tag{E.2}$$

Proof for Theorem.

An elegant proof can be found in [16]. Let's consider a discretized version of the Brownian motion, ie a random walk:

$$X_n = X_{n-1} + \xi_n \tag{E.3}$$
$$\xi_n \underset{[i.i.d.]}{\sim} \mathcal{N}(0, 1) \tag{E.4}$$

Then, $\forall \alpha_1, \alpha_2, ..., \alpha_n \in \mathbb{R}$,

$$\alpha_1 X_1 + \alpha_2 X_2 + ... + \alpha_n X_n = \alpha_1 \xi_1 + \alpha_2(\xi_1 + \xi_2) + ... + \alpha_n(\xi_1 + \xi_2 + ...\xi_n) \tag{E.5}$$
$$= (\alpha_1 + \alpha_2 + ... + \alpha_n)\xi_1 + (\alpha_2 + ... + \alpha_n)\xi_2 + ... + \alpha_n \xi_n \tag{E.6}$$

which is Gaussian as a linear combination of independent Gaussians. Therefore $X_{[1:n]} = (X_1, ..., X_n)$ is a Gaussian vector and $(X_n)_{n\geq 1}$ is a Gaussian process.

The law of the vector $X_{[1:n]}$ is computed using the caracteristic function (with $\alpha = (\alpha_1, ..., \alpha_n)$)

$$\begin{aligned}
\phi(\alpha) &= \mathbb{E}(e^{<\alpha, X_{[1:n]}>}) && \text{(E.7)}\\
&= \mathbb{E}(e^{((\alpha_1+\alpha_2+...+\alpha_n)\xi_1+(\alpha_2+...+\alpha_n)\xi_2+...+\alpha_n\xi_n)}) && \text{(E.8)}\\
&= \mathbb{E}(e^{(\alpha_1+\alpha_2+...+\alpha_n)\xi_1} e^{(\alpha_2+...+\alpha_n)\xi_2} ... e^{\alpha_n\xi_n}) && \text{(E.9)}\\
&= \mathbb{E}(e^{(\alpha_1+\alpha_2+...+\alpha_n)\xi_1})\mathbb{E}(e^{(\alpha_2+...+\alpha_n)\xi_2})...\mathbb{E}(e^{\alpha_n\xi_n}) && \text{(E.10)}\\
&= e^{-\frac{1}{2}(\alpha_1+\alpha_2+...+\alpha_n)^2} e^{-\frac{1}{2}(\alpha_2+...+\alpha_n)^2} ... e^{-\frac{1}{2}\alpha_n^2} && \text{(E.11)}\\
&= e^{-\frac{1}{2}\sum_{i=1}^n (\sum_{j=i}^n \alpha_j)^2} && \text{(E.12)}\\
&= e^{-\frac{1}{2}\|B\alpha\|^2} && \text{(E.13)}\\
&= e^{-\frac{1}{2}<\alpha, B^T B\alpha>} && \text{(E.14)}
\end{aligned}$$

with the matrix:

$$B = \begin{pmatrix} 1 & 0 & 0 & ... & 0 \\ 1 & 1 & 0 & ... & 0 \\ \vdots & & & & \\ 1 & 1 & 1 & ... & 1 \end{pmatrix} \tag{E.15}$$

and the covariance matrix $\Gamma = B^T B$:

$$\Gamma = \begin{pmatrix} 1 & 1 & 1 & ... & 1 \\ 1 & 2 & 2 & ... & 2 \\ 1 & 2 & 3 & ... & 3 \\ \vdots & & & & \\ 1 & 2 & 3 & ... & n \end{pmatrix} \tag{E.16}$$

ie $\Gamma_{i,j} = \min(i, j)$ ■

Let's look now at three points x_1, x_2, x_3, taken at times $t_1 < t_2 < t_3$.

By definition of a Gaussian process, their joint probability is Gaussian:

$$\begin{aligned}
\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} &\sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \Gamma(t_1,t_1) & \Gamma(t_1,t_2) & \Gamma(t_1,t_3) \\ \Gamma(t_2,t_1) & \Gamma(t_2,t_2) & \Gamma(t_3,t_3) \\ \Gamma(t_3,t_1) & \Gamma(t_3,t_2) & \Gamma(t_3,t_3) \end{pmatrix}\right) && \text{(E.17)}\\
&= \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} t_1 & t_1 & t_1 \\ t_1 & t_2 & t_2 \\ t_1 & t_2 & t_3 \end{pmatrix}\right) && \text{(E.18)}
\end{aligned}$$

We compute now $p(x_3|x_2, x_1)$ with Gaussian conditionning:

$$p(x_3|x_2, x_1) = \mathcal{N}(m_{3,12}, k_{3,12}) \tag{E.19}$$

$$m_{3,12} = \begin{pmatrix} t_1 & t_2 \end{pmatrix} \begin{pmatrix} t_1 & t_1 \\ t_1 & t_2 \end{pmatrix}^{-1} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{E.20}$$

$$k_{3,12} = t_3 - \begin{pmatrix} t_1 & t_2 \end{pmatrix} \begin{pmatrix} t_1 & t_1 \\ t_1 & t_2 \end{pmatrix}^{-1} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \tag{E.21}$$

We compute

$$\begin{pmatrix} t_1 & t_1 \\ t_1 & t_2 \end{pmatrix}^{-1} = \frac{1}{t_1(t_2 - t_1)} \begin{pmatrix} t_2 & -t_1 \\ -t_1 & t_1 \end{pmatrix} \tag{E.22}$$

and finally get:

$$m_{3,12} = x_2 \tag{E.23}$$

$$k_{3,12} = t_3 - t_2 \tag{E.24}$$

which proves that $p(x_3|x_2, x_1) = p(x_3|x_2) = \mathcal{N}(x_2, t_3 - t_2)$, ie the Browian motion is a Markov process, and we also retrieve that $x_3 - x_2 \sim \mathcal{N}(0, t_3 - t_2)$.

F

Adjoint sensitivity method

NB : I have chosen here to write vector with an underline : $\underline{u}$ versus a scalar u. Event though the notation is heavier, this should prove useful when coding the algorithm and checking for shapes of gradients and Jacobians.

In a machine learning setting, we model a variable $\underline{u}$ -usually a latent variable- as a vector-valued function verifying an ODE:

$$\frac{d\underline{u}}{dt} = f_\theta(\underline{u}, t) \tag{F.1}$$

$$\underline{u}(t = 0) = \underline{u}_0 \tag{F.2}$$

- Notations:
  - x is a scalar value
  - $\underline{x}$ is a vector
- t is the time.
- $\underline{u} : \mathbb{R} \to \mathbb{R}^N$ is a function of time into a space of dimension N.
- $f_\theta : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$ is a function parameterized by $\underline{\theta} \in \mathbb{R}^P$ - typically a neural network which we want to train. NB : f_θ can also be seen as a function $f(\underline{u}, t, \underline{\theta})$.

We wish to optimize a loss J function of the **function** $\underline{u}$ and parameters set $\underline{\theta}$:

$$J : (\mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N) \times \mathbb{R}^P \to \mathbb{R} \tag{F.3}$$

$$\underline{u}, \underline{\theta} \mapsto J(\underline{u}, \underline{\theta}) = \int_0^T g(\underline{u}, \underline{\theta}) dt \tag{F.4}$$

where g is a scalar-valued functional.

For example, we can set $g(\underline{u}, \underline{\theta}) = \underline{u}^T Q \underline{u}$, where Q is a $N \times N$ definite positive symetric matrix to compute a L_2 loss.

In order to train our neural network f_θ, we need to compute the gradients

$$\frac{dJ}{d\underline{\theta}} = \frac{d}{d\underline{\theta}} \int_0^T g(\underline{u}, \theta) dt \tag{F.5}$$

so we can backpropagate them.

Let's start the computation.

$$\frac{dJ}{d\underline{\theta}} = \frac{d}{d\underline{\theta}} \int_0^T g(\underline{u}, \underline{\theta}) dt \tag{F.6}$$

$$= \int_0^T \left(\frac{\partial g}{\partial \underline{\theta}} + \frac{\partial g}{\partial \underline{u}} \frac{d\underline{u}}{d\underline{\theta}} \right) dt \tag{F.7}$$

where $\frac{\partial g}{\partial \underline{\theta}}$ is a gradient (shape $1 \times P$ as a row), $\frac{\partial g}{\partial \underline{u}}$ is a gradient (shape $1 \times N$), and $\frac{d\underline{u}}{d\underline{\theta}}$ is a Jacobian $N \times P$.

The **adjoint method** consists in framing the problem as an optimization problem, and using the dual:

$$\min_{\underline{\theta}} : J(\underline{u}, \underline{\theta}) \tag{F.8}$$

$$\text{s.t } \frac{d\underline{u}}{dt} = f_\theta(\underline{u}, t) \tag{F.9}$$

$$\tag{F.10}$$

We form the Lagrangien of the problem, and compute its derivative wrt $\underline{\theta}$ (NB : the Lagrange multiplier λ is a function here):

$$\mathcal{L}(\underline{u}, \underline{\theta}, \underline{\lambda}) = \int_0^T \left[g(\underline{u}, \underline{\theta}) + \lambda^T(t) \left(f_\theta(\underline{u}, t) - \frac{d\underline{u}}{dt} \right) \right] dt \tag{F.11}$$

$$\frac{d\mathcal{L}}{d\underline{\theta}} = \int_0^T \left[\frac{\partial g}{\partial \underline{u}} \frac{d\underline{u}}{d\underline{\theta}} + \frac{\partial g}{\partial \underline{\theta}} + \lambda^T(t) \left(\frac{\partial f}{\partial \underline{u}} \frac{d\underline{u}}{d\underline{\theta}} + \frac{\partial f}{\partial \underline{\theta}} - \frac{d}{d\underline{\theta}} \frac{d\underline{u}}{dt} \right) \right] dt \tag{F.12}$$

We integrate by parts the problematic part of the integral:

$$\int_0^T \lambda^T(t) \frac{d}{d\underline{\theta}} \frac{d\underline{u}}{dt} dt = \int_0^T \lambda^T(t) \frac{d}{dt} \frac{d\underline{u}}{d\underline{\theta}} dt \tag{F.13}$$

$$= \lambda^T(t) \frac{d\underline{u}}{d\underline{\theta}} |_0^T - \int_0^T \left(\frac{d\lambda}{dt} \right)^T \frac{d\underline{u}}{d\underline{\theta}} dt \tag{F.14}$$

Finally:

$$\frac{d\mathcal{L}}{d\underline{\theta}} = \int_0^T \left[\left(\frac{\partial g}{\partial \underline{u}} + \left(\frac{d\lambda}{dt} \right)^T + \lambda^T(t) \frac{\partial f}{\partial \underline{u}} \right) \frac{d\underline{u}}{d\underline{\theta}} + \frac{\partial g}{\partial \underline{\theta}} + \lambda^T(t) \frac{\partial f}{\partial \underline{\theta}} \right] dt - \lambda^T(t) \frac{d\underline{u}}{d\underline{\theta}} |_0^T \tag{F.15}$$

Which leads to, canceling the factor of $\frac{d\underline{u}}{d\underline{\theta}}$ in the integral, and choosing $\lambda(T) = 0$:

$$\frac{d\lambda}{dt} + \left(\frac{\partial f}{\partial \underline{u}} \right)^T \lambda(t) + \left(\frac{\partial g}{\partial \underline{u}} \right)^T = 0 \tag{F.16}$$

$$\lambda(T) = 0 \tag{F.17}$$

$$\frac{d\mathcal{L}}{d\underline{\theta}} = \int_0^T \left(\frac{\partial g}{\partial \underline{\theta}} + \lambda^T(t) \frac{\partial f}{\partial \underline{\theta}} \right) dt + \lambda(0) \frac{d\underline{u}}{d\underline{\theta}} |_0 \tag{F.18}$$

Now, we remark that the solution $\underline{u}, \underline{\theta}$ is feasible, so $\frac{d\mathcal{L}}{d\underline{\theta}} = \frac{dJ}{d\underline{\theta}}$ as the ODE is verified.

Finally, the gradient of the loss wrt the parameter set $\underline{\theta}$ is given by:

Gradient computation by adjoint method

$$\frac{d\lambda}{dt} + \left(\frac{\partial \underline{f}}{\partial \underline{u}}\right)^T \lambda(t) + \left(\frac{\partial g}{\partial \underline{u}}\right)^T = 0 \tag{F.19}$$

$$\lambda(T) = 0 \tag{F.20}$$

$$\frac{dJ}{d\underline{\theta}} = \int_0^T \left(\frac{\partial g}{\partial \underline{\theta}} + \lambda^T(t)\frac{\partial \underline{f}}{\partial \underline{\theta}}\right) dt + \lambda(0)\frac{d\underline{u}_0}{d\underline{\theta}} \tag{F.21}$$

The Lagrange multiplier function λ is given by solving a terminal value ODE.

The overall strategy for computing the gradient of the loss J wrt to the parameter set $\underline{\theta}$ is therefore:

Strategy for computing the gradient of the loss J wrt $\underline{\theta}$

1. solve the **initial value problem** with an ODE solver:

$$\frac{d\underline{u}}{dt} = f_\theta(\underline{u}, t) \tag{F.22}$$

$$\underline{u}(t=0) = \underline{u}_0 \tag{F.23}$$

2. solve the **adjoint/terminal value problem** with an ODE solver:

$$\frac{d\lambda}{dt} + \left(\frac{\partial \underline{f}}{\partial \underline{u}}\right)^T \lambda(t) + \left(\frac{\partial g}{\partial \underline{u}}\right)^T = 0 \tag{F.24}$$

$$\lambda(T) = 0 \tag{F.25}$$

3. compute the gradient:

$$\frac{dJ}{d\underline{\theta}} = \int_0^T \left(\frac{\partial g}{\partial \underline{\theta}} + \lambda^T(t)\frac{\partial \underline{f}}{\partial \underline{\theta}}\right) dt + \lambda(0)\frac{d\underline{u}_0}{d\underline{\theta}} \tag{F.26}$$

where the derivatives $\frac{\partial g}{\partial \underline{\theta}}$, $\frac{\partial \underline{f}}{\partial \underline{\theta}}$, $\frac{\partial \underline{f}}{\partial \underline{u}}$ can be computed via automatic differentiation.

Glossary

ApEn Approximate Entropy. 78, 80

CD-SSM Continuous-Discrete State Space Model. 50–52

CT-SSM Continuous-Time State Space Model. 50, 51

DAG Directed Acyclic Graph. 11, 13, 17, 25

DKF Deep Kalman Filter. 7, 30, 31, 57

DVAE Dynamical Variational Auto Encoder. 7, 10, 14, 16, 17, 21, 24, 25, 41, 50–52, 57

ELBO Evidence Lower Bound. 18, 30, 86

GP Gaussian Process. 7, 10, 24–26, 36, 53, 54, 57

GP-VAE Gaussian Process Variational Auto Encoder. 7, 24, 36, 51, 53, 57

GPM Graphical Probabilistic Model. 10, 21, 24, 25

IT Information Theory. 78

LSTM Long Short Term Memory. 18, 22

MLP Multi Layer Perceptron. 10, 18

Neural-ODE Neural Ordinary Differential Equation. 7, 8, 57, 82

Neural-SDE Neural Stochastic Differential Equation. 7, 57

ODE Ordinary Differential Equation. 8, 46, 82, 83, 98–100

RTS Rauch-Tung-Striebel. 7, 52, 53

SampEn Sample Entropy. 78, 80

SDE Stochastic Differential Equation. 7, 24, 41, 44, 46–55, 57, 83, 91, 95

SSM State Space Model. 7

VAE Variational Auto Encoder. 7, 14–16, 24, 57

VLB Variational Lower Bound. 15, 16, 18, 21, 22, 26, 27, 83, 86

VRNN Variational Recurrent Neural Network. 7, 10, 21, 31, 33, 36, 57

Bibliography

[1] C Bishop. *Pattern Recognition and Machine Learning*. Accessed on Month Day, Year. 2006. URL: https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf.

[2] Francesco Paolo Casale et al. *Gaussian Process Prior Variational Autoencoders*. en. Oct. 2018. URL: https://arxiv.org/abs/1810.11738v2 (visited on 06/07/2025).

[3] Ricky T. Q. Chen et al. *Neural Ordinary Differential Equations*. arXiv:1806.07366. Dec. 2019. DOI: 10.48550/arXiv.1806.07366. URL: http://arxiv.org/abs/1806.07366 (visited on 09/05/2025).

[4] Junyoung Chung et al. *A Recurrent Latent Variable Model for Sequential Data*. arXiv:1506.02216. Apr. 2016. DOI: 10.48550/arXiv.1506.02216. URL: http://arxiv.org/abs/1506.02216 (visited on 06/07/2025).

[5] *Course materials: (Slides) - Machine learning with kernel methods / Spring 2025*. URL: https://mva-kernel-methods.github.io/course-page/ (visited on 08/11/2025).

[6] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition*. English. Hoboken, N.J: Wiley-Interscience, 2006. ISBN: 9780471241959.

[7] Alfonso Delgado-Bonal and Alexander Marshak. "Approximate Entropy and Sample Entropy: A Comprehensive Tutorial". en. In: *Entropy* 21.6 (June 2019), p. 541. ISSN: 1099-4300. DOI: 10.3390/e21060541. URL: https://www.mdpi.com/1099-4300/21/6/541 (visited on 01/20/2025).

[8] Vincent Fortuin et al. *GP-VAE: Deep Probabilistic Time Series Imputation*. arXiv:1907.04155. Feb. 2020. DOI: 10.48550/arXiv.1907.04155. URL: http://arxiv.org/abs/1907.04155 (visited on 06/07/2025).

[9] *Gaussian Processes for Machine Learning: Book webpage*. URL: https://gaussianprocess.org/gpml/ (visited on 06/07/2025).

[10] Laurent Girin et al. *Dynamical Variational Autoencoders: A Comprehensive Review*. arXiv:2008.12595. July 2022. DOI: 10.48550/arXiv.2008.12595. URL: http://arxiv.org/abs/2008.12595 (visited on 01/19/2025).

[11] Diederik P. Kingma and Max Welling. "An Introduction to Variational Autoencoders". In: (2019). DOI: 10.48550/ARXIV.1906.02691. URL: https://arxiv.org/abs/1906.02691 (visited on 07/21/2025).

[12] Diederik P. Kingma and Max Welling. *Auto-Encoding Variational Bayes*. arXiv:1312.6114. Dec. 2022. DOI: 10.48550/arXiv.1312.6114. URL: http://arxiv.org/abs/1312.6114 (visited on 08/08/2025).

[13] Friedman Koller. *Probabilistic Graphical Models*. en-US. URL: https://mitpress.mit.edu/9780262013192/probabilistic-graphical-models/ (visited on 07/21/2025).

[14] Xuechen Li et al. *Scalable Gradients for Stochastic Differential Equations*. arXiv:2001.01328. Oct. 2020. DOI: 10.48550/arXiv.2001.01328. URL: http://arxiv.org/abs/2001.01328 (visited on 09/05/2025).

[15] Yingzhen Li and Stephan Mandt. *Disentangled Sequential Autoencoder*. arXiv:1803.02991. June 2018. DOI: 10.48550/arXiv.1803.02991. URL: http://arxiv.org/abs/1803.02991 (visited on 07/23/2025).

[16] *Mouvement brownien et calcul d'Itô-Léonard Gallardo-Editions Hermann*. Sept. 2008. URL: https://www.editions-hermann.fr/livre/mouvement-brownien-et-calcul-d-ito-leonard-gallardo (visited on 04/16/2025).

[17] K Murphy. *Probabilistic Macine Learning Advanced Topics*. 2023. URL: https://mitpress.mit.edu/9780262048439/probabilistic-machine-learning/.

[18] *Page Web de Jean-François Le Gall*. URL: https://www.imo.universite-paris-saclay.fr/~jean-francois.le-gall/ (visited on 04/16/2025).

[19] Stefano Peluchetti and Stefano Favaro. *Infinitely deep neural networks as diffusion processes*. arXiv:1905.11065. Mar. 2020. DOI: 10.48550/arXiv.1905.11065. URL: http://arxiv.org/abs/1905.11065 (visited on 09/05/2025).

[20] S M Pincus. "Approximate entropy as a measure of system complexity." en. In: *Proceedings of the National Academy of Sciences* 88.6 (Mar. 1991), pp. 2297–2301. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.88.6.2297. URL: https://pnas.org/doi/full/10.1073/pnas.88.6.2297 (visited on 01/20/2025).

[21] L S. Pontriagin et al. *The mathematical theory of optimal processes*. eng. Ed. by Lucien W. Neustadt. Classics of Soviet mathematics. OCLC: 1035389999. Boca Raton: CRC Press, 2018. ISBN: 9780203749319.

[22] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. eng. 3. print. Adaptive computation and machine learning. Cambridge, Mass.: MIT Press, 2008. ISBN: 9780262182539.

[23] S. Roberts et al. "Gaussian processes for time-series modelling". en. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1984 (Feb. 2013), p. 20110550. ISSN: 1364-503X, 1471-2962. DOI: 10.1098/rsta.2011.0550. URL: https://royalsocietypublishing.org/doi/10.1098/rsta.2011.0550 (visited on 07/23/2025).

[24] Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. *Latent ODEs for Irregularly-Sampled Time Series*. arXiv:1907.03907. July 2019. DOI: 10.48550/arXiv.1907.03907. URL: http://arxiv.org/abs/1907.03907 (visited on 09/05/2025).

[25] Simo Särkkä and Arno Solin. *Applied Stochastic Differential Equations*. Institute of Mathematical Statistics Textbooks. Cambridge: Cambridge University Press, 2019. ISBN: 9781316510087. DOI: 10.1017/9781108186735. URL: https://www.cambridge.org/core/books/applied-stochastic-differential-equations/6BB1B8B0819F8C12616E4A0C78C29EAA (visited on 07/23/2025).

[26] C. E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3 (July 1948), pp. 379–423. ISSN: 0005-8580. DOI: 10.1002/j.1538-7305.1948.tb01338.x. URL: https://ieeexplore.ieee.org/document/6773024 (visited on 07/28/2025).

[27] Michalis Titsias and Neil D. Lawrence. "Bayesian Gaussian Process Latent Variable Model". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, pp. 844–851. URL: https://proceedings.mlr.press/v9/titsias10a.html.

[28] Harrison Zhu, Carles Balsells Rodas, and Yingzhen Li. *Markovian Gaussian Process Variational Autoencoders*. arXiv:2207.05543. Aug. 2023. DOI: 10.48550/arXiv.2207.05543. URL: http://arxiv.org/abs/2207.05543 (visited on 07/23/2025).