RIKEN AIP
Approximate Bayesian Inference Team

2018 - 2019

OREISTEIN Pierre

Engineering Student,
Applied Mathematics and Computer Sciences Department (AMCS)

Efficient Bayesian Deep Learning

(Apprentissage Profond Bayésien et Efficace)

Riken AIP: Approximate Bayesian Inference Team
Work lead in 2019 (May to September), in Riken AIP,
〒103-0027 Tokyo, Chuo City, Nihonbashi,
1 Chome一4一1 日本橋一丁目三井ビルディング15階

Academic tutor: M. Mohammad Emtiyaz KHAN,
Full Professor RIKEN AIP

Summary Sheet

- Internship Type: Projet de Fin d'Études (PFE)
- Duration: 18 weeks $\approx$ 4 months and a half
- Year: 2019
- Author: OREISTEIN Pierre
- Department: Applied Mathematics and Computer Sciences (AMCS)
- Title: Efficient Bayesian Deep Learning
- Titre: Apprentissage Profond Bayésien et Efficace
- Company: Riken, Advanced Center for Intelligence Project (AIP)
- Host Country: Japan
- Supervisors: Mohammad Emtiyaz KHAN
- Key words: Approximate Bayesian Inference, Functional Variational Inference, Sparse Gaussian Processes, Taylor Expansion

Acknowledgements

First of all, I would like to express my sincere acknowledgments to Ecole Nationale des Ponts ParisTech and all the staff of the IMI department, particularly **Ms. Sandrine GUILLERM**, **M. Mohammed El Rhabi** and **Ms. Alice TRAN** who provided us precious advice and support all along with my studies.

For the excellent management, the formidable quality of the courses and the availability of the teachers and personnel, I would like to formulate all my recognition to the team of the MVA master and in particular **Ms. Delphine LAVERNE**, **M. Nicolas VAYATIS** and **Ms. Agnès DESOLNEUX**.

For his extremely warm welcome, his precious feedback and his guidance, I would like to address all my gratitude to my supervisor at Riken, **M. Mohammad Emtiyaz KHAN**, the project leader of the Approximate Bayesian Inference Team. For his enthusiasm and his helpful advice, I would like to express all my thanks to my colleague **M. Alex IMMER**. For her constant help and her precious insights about Japan, I am also deeply grateful to **Ms. Harumi SEO**. For our instructive discussions, I also would like to mention **M. Mehdi ABBANA BENNANI** and **M. Vincent TAN** and all my other colleagues who allowed me to live such a rich experience.

Finally, I would like to thank my academic supervisors **Ms. Stéphanie ALLASSONNIÈRE** and **M. Pascal MONASSE** for their continuous backing.

Abstract

For my final internship, I had the immense privilege to join Riken AIP: one of the most prestigious research institute in Japan. During the almost 5 months of my internship, I had the opportunity to work and learn around the fascinating theory of Approximate Bayesian Inference. In particular, I had the chance to do research extensively on one of the newest and most fascinating algorithms developed by my welcoming team: VON for Variational Online Newton [Khan et al., 2018]. This algorithm combines all the qualities such as the efficiency and the scalability. Therefore, inspired by this algorithm, I tried to developed potential extension and, in particular, the infinite-dimensional case. Finally, I also dedicated my time to derive some analytic properties of VON that could explain its efficiency in prediction in practice.

Contents

List of Figures

Chapter 1

Introduction

First of all, I would like to thank once again all the people who gave me this formidable opportunity to work on such a fruitful topic and to earn a so rich and important experience. By the discovery of the marvelous Japanese culture, by the unexpected encounters and by the adventures I lived, I learned far beyond my work.

1.1 Choice of the internship

The final internship represents a formidable opportunity to deepen our knowledge on a precise topic. At master MVA, I particularly enjoyed the courses of M. Francis BACH on Probabilistic Graphical Models and the course of my supervisor Ms. Stéphanie ALLASSONNIÈRE on Computational Statistics. Indeed, I am a regular follower of the Youtube channel *Sciences for All* which made me discover the beauty of the Bayes formula and the illuminating courses of M. Stanislas DEHAENE at Collége de France. Therefore, I developed a personal attraction for Bayesian methods. When MVA published the offer of Riken and the opportunities to join the Approximate Bayesian Inference team, I do not hesitate for a second. My appeal for the Bayesian world combined with the possibility to discover the far-East culture of Japan perfectly matched all my aspirations.

1.2 Context and Riken Presentation

1.2.1 Presentation of Riken AIP and its activities

For my final internship, I had the privilege to join Riken AIP (Advance Center for Intelligence Project). Riken is one of Japan's oldest and largest fundamental-research institute. Nevertheless, Riken AIP is the youngest entity of Riken; it was just launched in April 2016 with the goal: "to achieve a scientific breakthrough and to contribute to the welfare of society and humanity through developing innovative technologies". Besides fundamental research, they "also conduct research on ethical, legal and social issues caused by the spread of AI technology and develop human resources." (cf Riken AIP Webpage)

1.2.2 Organisation of Riken AIP

Riken AIP is a center that clusters more than 50 teams and units from all over Japan. All these teams are divided into three groups:

- *Artificial Intelligence in Society Research Group.* The teams and units of this group mostly focused on understanding better the impact of artificial intelligence on the society from an ethical point of view. They aim to design law and technology that protects individuals. For example, the *Information Law Team* led by Masatomo SUZUKI (Ph.D.) focuses on the privacy and protection of medical information from a legal perspective (link).

(a) RIKEN Logo (b) AIP Logo

Figure 1.1 – Logos

- *Goal-Oriented Technology Research Group.* This group do research on very precise tasks using artificial intelligence tools. For example, the *Disaster Resilience Science Team* led by Naonori UEDA (D.Eng.) focuses on the prediction of earthquake occurrence for minimising the economic and social cost of such natural disasters (link).
- *Generic Technology Research Group.* This group has a more theoretical approach. The aim is to develop new techniques and algorithms of artificial intelligence with a general-purpose. For example, the *Tensor Learning Unit* led by Qibin Zhao (D.Eng.) aims to develop new algorithms to facilitate the learning from high-order structured data (link).

1.2.3 Presentation of the *Approximate Bayesian Inference Team*

In particular, I had the opportunity to join the team of Approximate Bayesian Inference (ABI) led by M. Mohammad Emtiyaz KHAN (Ph.D.) belonging to the *Generic Technology Research Group.* The aim of the team is "to understand the principles of learning from data and use them to develop algorithms that can learn like living beings." (link). In practice, the main goal of these last few years is to develop new algorithms using approximate Bayesian inference for different areas of machine learning such as deep learning, reinforcement learning, active learning, computer vision, and many others.

The team is composed of almost 15 people, depending on the back and forth, with M. KHAN as the team leader, two postdocs, two research assistants and 7 interns.

1.3 Choice of the Projects

Many projects are always going on inside the team and M. KHAN offered us true freedom in the selection of our project. At the time of my arrival, three main projects were beginning:

1. Variational Inference for Continual Learning
2. Variational Inference for Active Learning
3. Functional Variational Inference

All these projects were related to two recent results published by the team:

- *Fast and Scalable Bayesian Deep Learning by Weight-Perturbation in Adam*, [Khan et al., 2018]
- *Approximate Inference Turns Deep Networks into Gaussian Processes*, [Khan et al., 2019]

The first article presents a new fast and scalable algorithm for doing variational inference: VON (Acronym of Variational Online Newton). We will present it in the next chapter. The second article presents how VON is equivalent to a Gaussian process. Thanks to these two precious results, the idea was to apply this new algorithm to different fields like Reinforcement Learning or to focus on the theoretical implication of VON in the function space. Because of my appetence with the

theory, I dedicated my time to derive new theoretical perspectives for VON. In particular, I had the opportunity to work on three projects that I will present in this report:

1. VON with Functional Regularisation
2. Functional VON
3. VON as Taylor Expansion

In the first part, we will present VON and its main qualities such that its efficiency and above all its scalability. In the second part, we will present a method to extend this algorithm to meaningful prior. In the third part, we will give the first draft for extending VON to the infinite-dimensional case. Indeed, as we will explain it, the infinite-dimensional case could be the answer for scalable training of Gaussian Processes. Finally, in the fourth part, we will focus again on the classic VON algorithm and derive some nice equivalences. In particular, we will wonder if it could use another probability with the same attractive properties than the Gaussian in practice.

Chapter 2

Variational Online Newton (VON)

In this chapter, we are going to present VON. VON is an efficient and scalable algorithm to solve Variational Inference (VI) problems. In the first part, we are going to give some motivations for studying VI problems. In the second part, we will describe VON and give its origins and most attractive properties.

2.1 Motivations

2.1.1 Deep Learning suffers from a lack of uncertainty estimate

Since few years, Deep Learning techniques have completely transformed the research in Machine Learning. Their incredible expressive power combined with their ease of training made them an almost universal tool. Nevertheless, the utilisation of deep neural networks for practical application suffers from two main drawbacks:

1. Lack of interpretability
2. Lack of uncertainty measure on the prediction

Indeed, for the modelisation in the economy, the interpretation is a key component of a predictive model for fairness and justice. In the same vein, the recognition of malignant tumours in cancerology needs reliable uncertainty measure on the predictions to avoid all risk on patient health.

Solving these two issues are difficult as a result of the high non-linearity of deep neural networks and their keep-growing size. However, Bayesian methods and in particular Bayesian Neural Networks represent a promising solution for obtaining an estimate of the uncertainty around the predictions of a deep neural network.

2.1.2 Bayesian Modeling as a promising solution

Bayesian Inference offers an interesting framework with some advantages like:

- having a measure of the uncertainty around each prediction
- reducing the over-fitting. In fact, adding a prior on the parameters acts like a regulariser (cf section 2.3.2)
- Ability to generate data with the prediction law
- Automatic selection of a model (Ockham's razor). Choosing a prior is equivalent to promote some simpler models.

Indeed, in Bayesian modeling, it must model the two following distributions

1. *Likelihood:* $p((y, x) \mid w, \gamma)$, probability to observe the point (x, y) given the variable w
2. *Prior:* $p(w \mid \gamma)$, a priori knowledge about the probability to observe w

For example, w could be the weights of a linear regression. Therefore, $p(w \mid \gamma)$ could be a centred Gaussian with a diagonal covariance and $p((y, x) \mid w, \gamma)$ could be also a gaussian with mean $\sum_{i=1}^{d} w_i x_i$ and a diagonal covariance. Finally, γ could represent some hyperparameters like the variance of the noise associated with the linear regression.

Therefore, according to the Bayes rule it can compute

1. *The Posterior Predictive distribution:*

$$\begin{aligned} p((y, x) \mid D, \gamma) &= \int_{w \in \Omega} p(y \mid w, \gamma) p(w \mid D, \gamma) \mathrm{d}w \\ &= \int_{w \in \Omega} p(y \mid w, \gamma) \frac{p(D \mid w, \gamma) p(w \mid \gamma)}{p(D \mid \gamma)} \mathrm{d}w \end{aligned} \tag{2.1}$$

2. *The most likely model by maximising the marginal likelihood:*

$$\max_{\gamma} p(D \mid \gamma) = \int_{w \in \Omega} p(D \mid w, \gamma) p(w \mid \gamma) \mathrm{d}w \tag{2.2}$$

In other words, we succeed to

1. Modelise the posterior predictive distribution $p((y, x) \mid D, \gamma)$ and therefore to obtain an estimate of the uncertainty around a predictive sample.
2. To be able to generate new data point thanks to $p((y, x) \mid w, \gamma)$ and $p(w \mid D, \gamma)$.

However, to compute the first quantity, it applied the Bayes rule on the *posetrior* $p(w \mid D, \gamma)$ and in practice, applying the Bayes rule could be a very difficult task.

2.1.3 Difficulties to apply the Bayes rule

Applying the Bayes rule implies to be able to compute the normalising constant $p(D \mid \gamma)$, also called *marginal-likelihood*. For recall, it has:

$$\underbrace{p(w \mid D, \gamma)}_{\textit{posterior}} = \frac{p(w, D \mid \gamma)}{p(D \mid \gamma)} = \frac{\overbrace{p(D \mid w, \gamma)}^{\textit{likelihood}} \overbrace{p(w \mid \gamma)}^{\textit{prior}}}{\underbrace{p(D \mid \gamma)}_{\textit{normalising constant}}} \tag{2.3}$$

The *likelihood* and *prior* are choices of modelisation. Therefore, they are known. However, it still remains to compute the normalising constant $p(D \mid \gamma)$. In this situation, two cases are possible:

- First, it has a closed-form solution (see Appendix D.4.1) for the *normalising constant*. Basically, it corresponds to cases where the product of the *likelihood* with the *prior* is equal to a known distribution like Gaussian or Gamma. Therefore, it has a closed-form solution for the normalising constant. To be precise, such cases correspond to distributions which are conjugate. See Appendix B.1 for more explanations on the conjugacy.
- Second, it does not know how a close-form solution of the *normalising constant*. Therefore other techniques need to be deployed as we are going to see in the next sections.

As the first case does not need further study, it is going to focus on the second case. Basically, it wants to approximate the *normalising constant*. In practice, it is equal to the integral of the joint distribution over the parameters and latent variables w:

$$p(D \mid \gamma) = \int_{w \in \Omega} p(w, D \mid \gamma)\, dw = \int_{w \in \Omega} p(D \mid w, \gamma) p(w \mid \gamma)\, dw \tag{2.4}$$

A first idea could be to use MC methods like MCMC ones to approximate this integral. This could work in small-dimension. However, as it is going to work with deep neural networks, the dimension of the space of latent variables/parameters w could be superior to several million or even billions. Therefore, MC methods are not more reliable, and other techniques have to be developed.

Instead of trying to approximate the *normalising constant*, it can try to approximate directly the *posterior* with a distribution of a well-known family like Gaussians. The advantage of this technique is the guarantee to obtain a true distribution after approximation. Today, three main techniques exist using this idea:

1. Laplace's method
2. Expectation propagation of [Hernández-Lobato and Adams, 2015]
3. Variational Inference firstly introduced by M. Jordan.

For simplicity, we will just describe the first and last method in this report.

2.2 Laplace Method

The Laplace's method consists in approximating the *posterior* with a Gaussian. The choice of the parameters of the approximating Gaussian is quite simple:

- The mean w^* corresponds to a local maximum of the *log-likelihood*: $\ell(w) \coloneqq \ln p(D \mid w, \gamma)$
- The covariance matrix corresponds to the Hessian of the *log-likelihood* in w^*: $\boldsymbol{H}_w \ell(w^*)$

Therefore, it has to solve:

$$\nabla_w \ln p(D \mid w^*, \gamma) = 0 \quad \text{and} \quad \Sigma = \boldsymbol{H}_w \ell(w^*) = \nabla_w^2 \ln p(D \mid w^*, \gamma) \tag{2.5}$$

It works well in practice when both the *prior* and the *likelihood* are Gaussians but becomes limited in the other situations.

2.3 Variational Inference (VI)

2.3.1 Minimisation of the Kullback-Lieber divergence

This technique consists in approximating the *posterior* distribution $p(w \mid D, \gamma)$ with a parametrised distribution q_λ of a tractable family of distributions $\mathcal{P}$. For example, $\mathcal{P}$ could be the family of gaussians and in this case, λ could be the natural parameter: $\lambda = \left(\Sigma^{-1}\mu, \frac{-1}{2}\Sigma^{-1}\right)$ (cf section 2.4).

Once a choice of tractable family $\mathcal{P}$ is made, the *posterior* is approximated by minimising a distance between q_λ and the true *posterior*. In practice, the distance corresponds to the *Kullback-Lieber* divergence between q_λ and the *posterior*. Therefore, it tries to solve:

$$\min_{q_\lambda(w) \in \mathcal{P}} KL\left(q_\lambda \,||\, p(w \mid D, \gamma)\right) \tag{2.6}$$

This choice of minimisation problem makes sens according to the Information Theory of Shannon. For more details, please refer to the Appendix A.

2.3.2 Other formulations

To get more intuition about this criteria, it can rewrite it in few meaningful formulations. Let's describe some of them.

- First let's introduce a new objective function $\mathcal{L}_{VI}$

$$\mathcal{L}_{VI}(\lambda) = \ln p(D \mid \gamma) - KL\left(q_\lambda \,||\, p(w \mid D, \gamma)\right) \tag{2.7}$$

  As $\ln p(D \mid \gamma)$ is constant with respect to q_λ, maximising $\mathcal{L}_{VI}$ leads to the same argmax. This new formulation is interesting because $KL\left(q_\lambda \,||\, p(w \mid D, \gamma)\right) \geq 0$. Therefore, maximising $\mathcal{L}_{VI}$ gives $\ln p(D \mid \gamma)$. That is why $-\mathcal{L}_{VI}$ is often called the *Evident Lower Bound* (ELBO).

- It can also rewrite this new objective function $\mathcal{L}_{VI}$ as a problem of minimisation of a *Free-Energy*:

$$\mathcal{F}_{VI}(\lambda) = -\mathbb{E}_{q_\lambda}\left[\ln p(w, D \mid \gamma)\right] - H(q_\lambda) \tag{2.8}$$

  For recall, in physics, a free-energy measures the useful work obtainable from a closed thermodynamic system at a constant temperature and volume. Therefore, minimising this *Free-Energy* could be interpreted as minimising the number of irreversible transformations that describe the passage from the *posterior* to q_λ

- Finally, in the same way, $\mathcal{L}_{VI}$ can be rewritten as a more classic Machine Learning criteria with a loss and a regulariser:

$$-\mathcal{L}_{VI}(\lambda) = \underbrace{-\overbrace{\mathbb{E}_{q_\lambda}\left[\ln p(D \mid w, \gamma)\right]}^{\text{Data-fit term}}}_{\textit{loss}} + \underbrace{KL(q_\lambda \| \overbrace{p(w \mid D)}^{\textit{prior}})}_{\text{regulariser}} \tag{2.9}$$

  Therefore, minimising the objective leads to find the distribution which maximises the balance between maximising the *likelihood* and minimising the distance with the *prior*.

Now, it get a better understanding of this criteria, it remains to minimise it. First, it needs to select an approximate distribution $p_\lambda(w)$. As for VON [Khan et al., 2018], we are going to focus on exponential families for their nice properties. First, let us do some recalls on exponential families.

2.4 Exponential Families

2.4.1 Definition and Properties

Definition 2.4.1 – *Exponential Family*

Let us suppose

(H.1) p is a probability distribution (discrete or continuous)

Then, one says that p is a member of an exponential family if it can be factorised as

$$p(w \mid \gamma) = h(w) \exp\left(b(\gamma)^T \Phi(w) - \tilde{A}(\gamma)\right)$$

With:

- *$h(x)$ the ancillary statistic*
- *$h(x)d\mu(x)$ the reference measure (or base measure)*
- *$\Phi(x)$ the sufficient statistic (also called feature vector)*
- *γ the parameter*

- $\lambda = b(\gamma)$ *the canonical or natural parameter*
- $\tilde{A}(\gamma) = A(\lambda)$ *the log-partition function*

Remark 2.4.1 – *Sufficient Statistics*

It contains all the information of the effect of the parameter γ on the iid random variables X_i.
It can characterise a sufficient statistic by the theorem of factorisation: S(X) is a sufficient statistics if and only if

$$p(x \mid \gamma) = h(x) f(S(X), \gamma) \tag{2.10}$$

Remark 2.4.2 – *Role of the Ancillary Statistics*

h is useful for defining the non zero set. In fact, if $\forall w,\ h(w) \neq 0$, it can rewrite the previous formula without h by choosing $\tilde{\Phi}(w) = (\Phi_1(w), \cdots, \Phi_d(w),\ \ln h(w))^T$ and $\tilde{\lambda} = \left(\lambda^T, 1\right)^T$.

Properties 2.4.1 – *Minimal Representation and Convexity*

(P.1) p is canonical *or* natural *if and only if $b(\gamma) = \gamma = \lambda$*

(P.2) the domain of an exponential family is defined by $\Omega = \left\{\lambda \in \mathbb{R}^d \mid A(\lambda) < \infty\right\}$

(P.3) For recall, the vector of statistics is said to be affinely dependent:
If it denotes $\Phi(w) = (\Phi_1(w), \cdots, \Phi_d(w))^T$, it has:

$$\exists c \in \mathbb{R}^{d+1} \backslash \{0\}, \quad \forall w \in \{w \mid h(w) \neq 0\}, \quad c_0 + c_1 \Phi_1(w) + \cdots + c_d \Phi_d(w) = 0$$

One says that a vector of sufficient statistics Φ provides a minimal representation *of the exponential family if these statistics are affinely independent.*

(P.4) An exponential family is said to be curved *if its Jacobian $J = \left\{\frac{\partial b_i(\gamma)}{\partial \gamma_j}\right\}$ is not full rank. For example: $p_\lambda(w) = \mathcal{N}(w \mid \mu, \mu^2)$ is a curved exponential family.*

(P.5) Every exponential family admits at least one *minimal representation*

(P.6) Ω is a convex *subset of $\mathbb{R}^d$*

(P.7) $Z : \lambda \to \int \exp\left(\lambda^T \Phi(w)\right) h(w)\, dw$ is a convex *function*

(P.8) $A : \lambda \to \ln Z(\lambda)$ is a convex *function*

Theorem 2.4.1 – *Expectation Parameter-Natural Parameter Relationship*

Let us suppose

(P.1) p_λ is an exponential family not curved

(P.2) $\lambda \in \mathring{\Omega}$

Then,

$$\frac{\partial A(\lambda)}{\partial \lambda_k} = \mathbb{E}_{p_\lambda}\left[\Phi_k(X)\right]$$

$$\frac{\partial^m A(\lambda)}{\partial^{m_1}\lambda_1 \cdots \partial^{m_d}\lambda_d} = \mathbb{E}_{p_\lambda}\left[\Phi_1(X)^{m_1} \cdots \Phi_d(X)^{m_d}\right]$$

Corollary – Theorem 2.4.1

Let us suppose

(H.1) $q_\lambda := h(z)\exp\left(\Phi(z)^T\lambda - A(\lambda)\right)$ *is a member of an exponential family with an expectation parameter* $m(\lambda)$ *equal to* $m(\lambda) = \mathbb{E}_{q_\lambda}\left[\Phi(z)\right]$

Then,

1. $J_m\lambda = (\mathrm{Cov}[\Phi])^{-1} = F(\lambda)^{-1}$
2. $\nabla_m A(\lambda) = F(\lambda)^{-1}\mathbb{E}_{q_\lambda}\left[\Phi(z)\right]$

Where $J_m\lambda$ *is the Jacobian of* λ *according to* m *and* $F(\lambda)$ *is the Fisher information matrix of* q_λ.

The first equation can be proven by using the following equation $\nabla_\lambda A(\nabla_m A^*(m)) = m$. A^* is the Legendre Transform of A and this equality comes the convexity of A. The second equation can be proven using the chain rule.

2.5 Variational Inference with Exponential Families

Let us suppose that all the probabilities considered are a member of an exponential family. Then, it can rewrite the problem (2.6) as

$$\begin{aligned}\max_\lambda \mathcal{L}_{VI}(\lambda) = \max_\lambda \; \mathbb{E}_{q_\lambda}\Bigg[&\left(\langle \lambda_{lik}(w), \Phi_{lik}(D)\rangle_{lik} - A_{lik}(\lambda_{lik}(w))\right) \\ &+ \left(\langle \lambda_{prior}, \Phi_{prior}(w)\rangle_{prior} - A_{prior}(\lambda_{prior})\right) \\ &- \left(\langle \lambda, \Phi(w)\rangle - A(\lambda)\right)\Bigg]\end{aligned} \tag{2.11}$$

Where it denotes with

- lik the parameters and functions corresponding to the *likelihood*
- $prior$ the parameters and functions corresponding to the *prior*
- For the *posterior*, we skipped the subscript *post* for simplicity.

For sake of simplicity, it will assume that both the *prior* and the *posterior* belongs to the same family in the following. Therefore (2.11) becomes

$$\begin{aligned}&\max_\lambda \mathcal{L}_{VI}(\lambda) \\ &= \max_\lambda \; \mathbb{E}_{q_\lambda}\Bigg[\overbrace{\left(\langle \lambda_{lik}(w), \Phi_{lik}(D)\rangle_{lik} - A_{lik}(\lambda_{lik}(w))\right)}^{:=\ell(w)} - \left(\langle \lambda - \lambda_{prior}, \Phi(w)\rangle - A(\lambda) + A(\lambda_{prior})\right)\Bigg] \\ &= \max_\lambda \; \mathbb{E}_{q_\lambda}\left[\ell(w)\,\right] - KL\left(q_\lambda(w) \,||\, p(w)\right)\end{aligned} \tag{2.12}$$

Now, it remains to solve our minimisation problem (2.12).

2.6 Natural Gradient

2.6.1 Taking into account the Riemannian metric of the natural parameter

A classic way for minimising such problems would be to apply a classic gradient descent method according to the natural parameter. This gradient is easy to compute because of the linear part and the log-partition function (cf (2.12)). However, it will be sub-optimal. In fact, for many exponential family, the natural parameter does not belong to an Euclidean space but a Riemannian one. For example, the multivariate Gaussian distributions form an exponential family with

- *Sufficient statistics:* $\Phi(x) = \left(X, XX^T\right)^T \in \mathbb{R}^p \times Sym_+(\mathbb{R}^{p\times p})$
- *Natural Parameter:* $\lambda = \left(\mu^T\Sigma^{-1}, -\frac{1}{2}\Sigma^{-1}\right)^T \in \mathbb{R}^p \times Sym_+(\mathbb{R}^{p\times p})$
- *Log-partition Function:* $A(\eta) = \frac{1}{2}\mu^T\Sigma^{-1}\mu + \frac{N}{2}\ln 2\pi + \frac{1}{2}\ln|\Sigma|$

Therefore, it needs to take into account that our natural parameter lives in a Riemannian space.

Fortunately, [Shun-ichi Amari, 1998] proved that the metric associated with the natural parameter of an exponential family is easy to obtain. In fact, the metric is precisely equal to the Fischer matrix associated to the probability distribution: $F(\lambda)$. Therefore, instead of considering the classic gradient descent, it can takes into acount this metric such that our new optimisation scheme becomes:

$$\lambda_{t+1} = \lambda_t + \beta_t F(\lambda)^{-1}\nabla_\lambda \mathcal{L}_{VI} \tag{2.13}$$

For more details on how to obtain this equation, please refer to [Shun-ichi Amari, 1998].

Remark 2.6.1 – *Another motivation for using the natural gradient*

> *A final motivation to use the natural gradient is its convergence in one step in case of conjugacy. For more details, it could refer the appendix B.*

2.6.2 The convexity properties of Exponential Families allows simple updates

Nevertheless, as we are interested by training Bayesian neural networks, in other words a high-dimension case, inverting the Fischer matrix could be prohibitive. Fortunately, the convexity of the log-partition function associated with the exponential family offers a second fascinating relationship

$$F(\lambda)^{-1}\nabla_\lambda \mathcal{L}_{VI} = \nabla_m \mathcal{L}_{VI} \tag{2.14}$$

Where $m := \mathbb{E}_{q_\lambda}[\Phi(w)]$ denotes the expectation parameter.
For more details, please get look to the very well written paper of [Raskutti and Mukherjee, 2015].

Therefore, our optimisation scheme becomes

$$\lambda_{t+1} = \lambda_t + \beta_t \nabla_m \mathcal{L}_{VI} \tag{2.15}$$

A direct remark will be to question if computing the derivative according to m is still as simple as computing the derivative according to λ. Fortunately, it is easy to compute in many cases. Indeed, for any exponential family, the following property holds

Lemma 2.6.1

Let suppose

(H.1) p_{λ_0} *and* p_λ *are two distributions belonging to the same exponential family with respective natural parameters* λ_0*, fixed, and* λ

(H.2) Let us denote by $m := \mathbb{E}_{p_\lambda}[\Phi(X)]$ *the expectation parameter of* p_λ

Then,

$$\nabla_m KL\left(p_\lambda \,||\, p_{\lambda_0}\right) = -\nabla_m \mathbb{E}_{p_\lambda}\left[\ln \frac{p_{\lambda_0}(w)}{p_\lambda(w)}\right] = \lambda - \lambda_0$$

This equality can be proven with a direct calculus and by using the classic properties of exponential families (cf (2.4.1)).
Therefore, by using the Lemma (2.6.1) and (2.12), it obtains finally the VON update so desired (cf [Khan et al., 2018])

$$\lambda_{t+1} = (1-\beta_t) * \lambda_t + \beta_t * \left(\lambda_{prior} + \nabla_m \mathbb{E}_{q_\lambda}\left[\ell(w)\right]\right) \tag{2.16}$$

From this third transformation, it observes the first interesting property of VON. It corresponds to a moving average between the previous iteration and the new update. Therefore the natural gradient applied to a VI problem gives naturally a stable update equation. In particular, it will ensure a better convergence when we will approximate the expectation by a Monte-Carlo method.

Finally, it remains to compute $\nabla_m \mathbb{E}_{q_\lambda}\left[\ell(w)\right]$. This gradient of this term depends a lot on the choice of the approximate posterior. In practice, researchers often considers a Gaussian distribution for its simplicity. Therefore, in the next section, we will derive the VON equation for the Gaussian case and show that in such a case, it could obtain particularly attracting equations.

2.7 VON with a Gaussian Approximate Distribution

2.7.1 Bonnet s and Price s Theorems

If it uses a Gaussian distribution as approximated distribution, it could derive simpler updates thanks to the two theorems of Bonnet and Price.

Theorem 2.7.1 – *Bonnet's Theorem*

Let,

1. q *a multivariate Gaussian distribution with mean* μ *and covariance matrix* Σ
2. $f : \mathbb{R}^d \to \mathbb{R}$ *a multivariate function*

If,

(H.1) f *is a locally ACL function (cf D.3.2)*

(H.2) $\mathbb{E}_q\left[\left\|\Sigma^{-1}(x-\mu) f(x)\right\|\right] < +\infty$ *and* $\mathbb{E}_q\left[\|\nabla_x f(x)\|\right] < +\infty$

(H.3) The regular conditions for swapping of differentiation and integration are also satisfied

Then, the following first-order identity holds:

$$\frac{\partial}{\partial \mu}\mathbb{E}_q\left[f(x)\right] = \mathbb{E}_q\left[\Sigma^{-1}(x-\mu) f(x)\right] = \mathbb{E}_q\left[\nabla_x f(x)\right]$$

And,

Theorem 2.7.2 – *Price's Theorem*

Let,

1. q *a multivariate Gaussian distribution with mean* μ *and covariance matrix* Σ
2. $f : \mathbb{R}^d \to \mathbb{R}$ *a multivariate function*

If,

(H.1) f *is a continuously differentiable function.*

(H.2) $\nabla_x f$ *is a locally ACL function (cf D.3.2)*

(H.3) $\mathbb{E}_q\left[\left\|\nabla_x\left[\Sigma^{-1}\left(x-\mu\right)f(x)\right]\right\|\right] < +\infty$, $\mathbb{E}_q\left[\|\nabla_x^2 f(x)\|\right] < +\infty$ *and* $\mathbb{E}_q\left[|f(x)|\right] < +\infty$

(H.4) The regular conditions for swapping of differentiation and integration are also satisfied

Then, the following second-order identity holds:

$$\frac{\partial}{\partial\Sigma}\mathbb{E}_q\left[f(x)\right] = \frac{1}{2}\mathbb{E}_q\left[\Sigma^{-1}\left(x-\mu\right)\nabla_x^T f(x)\right] = \frac{1}{2}\mathbb{E}_q\left[\nabla_x^2 f(x)\right]$$

Thanks to these two theorems, it could apply the chain rule on m and these two theorems to finally obtain (cf [Khan et al., 2018])

$$\begin{cases} \boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t - \beta_t(\hat{\boldsymbol{g}}(w_t) + \tilde{\delta}\mu_t)/(s_{t+1} + \tilde{\delta}) \\ \boldsymbol{s}_{t+1} = (1-\beta_t) * \boldsymbol{s}_t + \beta_t diag\left[\boldsymbol{H}_w(w_t)\right] \end{cases} \tag{2.17}$$

With

- $\boldsymbol{a}/\boldsymbol{b}$ is an element-wise division operation between the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$
- $\hat{\boldsymbol{g}}(w_t)$ is a one sample estimate of the expectation the gradient of the loss: $\mathbb{E}_{q_{\lambda_t}}\left[\nabla_w \ell(w)\right]$
- $\hat{\boldsymbol{H}}(w_t)$ is a one sample estimate of the expectation the hessian of the loss: $\mathbb{E}_{q_{\lambda_t}}\left[\nabla_w^2 \ell(w)\right]$
- w_t is a sample of $\mathcal{N}\left(\mu_t, \sigma_t^2\right)$ with $\sigma_t^2 = 1/\left[N(\boldsymbol{s}_t + \tilde{\delta})\right]$
- $\tilde{\delta} \coloneqq \delta/N$ with δ the precision of the prior

For more details about the calculus, please refer to 5.2.2, for the details on the variables, please refer to [Khan et al., 2018].

These two simple equations are fascinating because from them it could easily derived all the most famous existing methods for training a deep neural networks such that

- Adam
- Adagrad
- RMSProp
- ...

2.7.2 VON: Properties

From (2.17), it deduces all the interesting properties of VON

- *Variational* because of its purpose to learn a the parameter of random variable
- *Online* because of its Bayesian origin it can handle online data
- *Newton* because the Fischer matrix is almost the true Hessian

However, as we are going to see in the next section, some aspects of VON can still be improved.

Chapter 3

Project I - VON with Functional Regularisation

As explained in the introduction, at my arrival, three main projects were going to start. Among the three, I get particularly interested by the wish of my supervisor to derive an equivalent of VON in the function-space. Indeed, the function-space offers two attractive properties

- Design of meaningful prior
- Scalable training of Gaussian Processes

In this chapter, we will develop a first algorithm inspired by VON using meaningful prior over functions. In the next chapter, we will design another algorithm for the scalable training of Gaussian Processes.

3.1 The Quest of a Meaningful Prior

In the weight-space, the classic Bayesian rule can be written as

$$\underbrace{q_\lambda(w)}_{\text{Approximate Posterior}} \approx \underbrace{p(w \mid D)}_{\text{Exact Posterior}} \propto \underbrace{p(D \mid w)}_{Likelihood}\underbrace{p(w)}_{Prior} \tag{3.1}$$

For example in [Khan et al., 2018], it assumes

- $p(D \mid w) = \exp(\ell(w))/Z$ with $\ell(w)$ the least-square error and Z the normalising constant
- $q_\lambda(w)$ is a multi-variate Gaussian.
- $p(w) = \mathcal{N}(0, \lambda^{-1} I)$

If it follows the classic interpretation of the different terms in the Bayes rule, the *prior* represents our a priori knowledge about the weights w. For example, in Bayesian Deep Learning w represents the weights of the neural networks. However, because of the high non-linearity of deep neural networks, having a priori knowledge of the weights is difficult. Therefore, the *prior* often equals to a multi-variate Gaussian for different practical reasons but not for interpretability reasons.

Instead, to avoid such a problem, we propose to do the Variational Inference directly in the function-space. In such a case, the Bayesian rule becomes

$$p(f \mid D) \propto p(D \mid w)p(f) \tag{3.2}$$

For example

- $p(f)$ could be equal to a Gaussian Process (GP) $\mathcal{N}(0, K(.,.))$ with K a kernel with some properties like periodicity or smoothness. For a brief introduction to Gaussian Processes, please refer to the Appendix C

Therefore, thanks for example to the GP, it becomes much easier to define meaningful *prior*. Indeed, it can now specify some precise properties directly on the functions thanks to the kernel of the GP like smoothness or periodicity as we are going to see in the next section.

3.2 Training a Bayesian Neural Network with meaningful prior: Some Experiments

Regularisation of the functions produced by a deep neural network is difficult if we work only with the weights. However, once it works in the function-space it is much easier to regularise the functions produced by using a GP *prior* with a certain kernel.

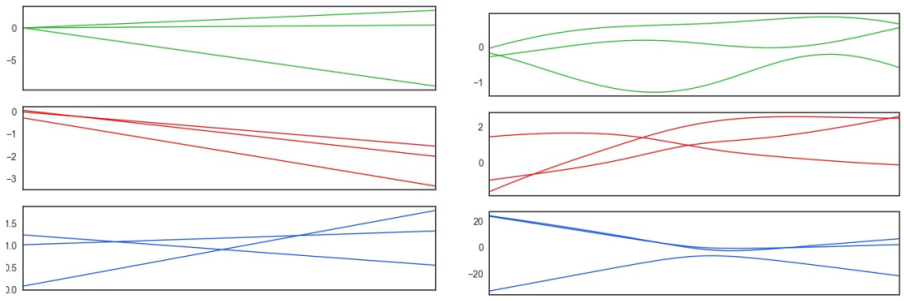
In 2017, [Flam-Shepherd et al., 2017] got the same idea and runs several experiments to check the possibility of such a method. Below, we present their results. We present in particular the results of two of their experiences. They trained a shallow Bayesian neural network (BNN) of 2 fully connected hidden layers by using VI and using a GP *prior*. In particular, they try to minimise

$$KL\left(p(f_{BNN}(X) \mid \Phi) \mid\mid p(D \mid f(X)) p_{GP}(f(X))\right) \tag{3.3}$$

With

- X the inputs inside our dataset D
- $p_{GP}(f(X)) = \mathcal{N}(0, K(.,.))$
- $\Phi = (\mu, \Sigma)$ such that $w \sim \mathcal{N}(\mu, \Phi)$
- $p_{BNN}(f(X))$ corresponds to the probability over the outputs of a neural networks. In practice, they used random weights to produce random outputs $f(X)$. They also do some approximations such that they just need to be able to sample from $p_{BNN}(f(X))$ but they do not need to have access to its density.

In a first experience, they train the neural network with a linear kernel and a smooth one



$k(x, x') = xx'$ $\quad$ $k(x, x') = e^{-(x-x')^2/50}$

$a(x) = x$ $\quad$ $a(x) = \log(1 + e^x)$

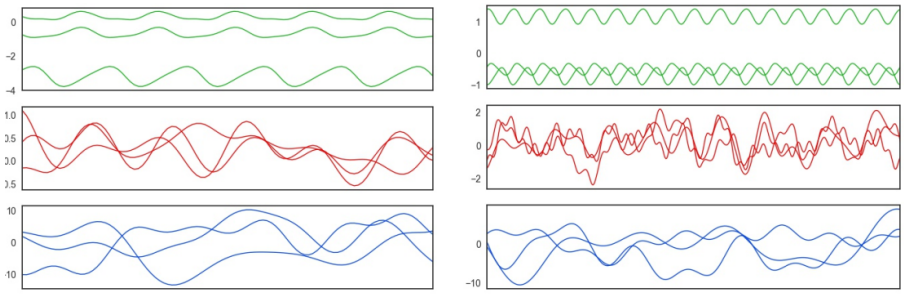
$$\boldsymbol{f(X) \sim p_{\text{GP}}(f(X))}, \quad \boldsymbol{f(X) \sim p_{\text{BNN}}(f(X)|\phi^*)}, \quad \boldsymbol{f(X) \sim p_{\text{BNN}}(f(X))}$$

Figure 3.1 – Training of a Bayesian Neural Networks with a linear kernel (right) and a smooth kernel (left). The function a corresponds to the activation function used inside the neural network. In green, samples of functions produced by the GP *prior*. In red, samples of functions produced by the trained Bayesian neural network and in blue, samples of functions produced by a non-trained Bayesian neural networks.

From this first experiment, we can observe the first promising result. On the right, it can observe that the trained Bayesian neural network (in red) succeed to produced almost linear functions like

the GP (in green) at the opposite of the non-trained one (in blue). In the same vein, on the left, it can observe smoother functions produced by the trained Bayesian neural network when they used a Gaussian kernel.

In a second experiment, they try to train a neural network to produce periodic functions. For doing so, they used, in particular, a periodic kernel with $\sin$ as the activation function.



$k(x, x') = e^{-2\sin^2(\pi|x-x'|)}$ $\quad$ $k(x, x') = e^{-2\sin^2(\pi|x-x'|)/25}$

$a(x) = \sin x$ $\quad$ $a(x) = \sin x$

$$\boldsymbol{f}(\mathbf{X}) \sim p_{\text{GP}}(\boldsymbol{f}(\boldsymbol{X})), \quad \boldsymbol{f}(\mathbf{X}) \sim p_{\text{BNN}}(\boldsymbol{f}(\boldsymbol{X})|\boldsymbol{\phi}^*), \quad \boldsymbol{f}(\mathbf{X}) \sim p_{\text{BNN}}(\boldsymbol{f}(\boldsymbol{X}))$$

Figure 3.2 – Training of a Bayesian Neural Networks with two different periodic kernels. The function a corresponds to the activation function used inside the neural network. In green, samples of functions produced by the GP *prior*. In red, samples of functions produced by the trained Bayesian neural network and in blue, samples of functions produced by a non-trained Bayesian neural networks.

Here, we can observe that the trained BNN tends to produce functions with a periodicity close to one of the functions produced by the GP.

Thanks to these first results, it gives us great hope that VI in the function-space is the very flexible and useful tool to train any BNN with a meaningful *prior* or regulariser. Therefore, the question becomes: how to adapt VON to do VI in the function-space such that it can use meaningful *prior* and be as efficient and stable as VON in the weight-space?

First, as for VON, it needs a relevant choice for the approximate posterior $q_\lambda(f(X))$. [Flam-Shepherd et al., 2017] uses strong approximations like neglecting the entropy term such they do not need the density of the approximative posterior for solving the VI problem but just samples. However, for applying VON, we would like to avoid these approximations and find therefore a relevant choice of approximative posterior for $q_\lambda(f(X))$.

3.3 Defining a Relevant Approximate Posterior

To be able to train a BNN in practice, the easiest way to do is to put a Gaussian distribution over the weights. Therefore, it could build upon such a distribution over the weights to obtain a distribution over the functions by a simple integration as follows

$$q_\lambda(f_X) = \int_{w\in\Omega} \mathbb{1}_{f_X = f(w,X)} q_\lambda(w) \mathrm{d}w \tag{3.4}$$

However, integrating and deriving an indicatrix is never easy in practice. Also, VON works well because it uses a distribution that belongs to an exponential family. Here, it is not the case.

Instead, we propose to approximate the indicatrix function by a multi-variate Gaussian

$$q_\lambda(f_X) = \int_{w\in\Omega} \mathcal{N}\left(f_X \mid f(w,X), \sigma^2 I\right) q_\lambda(w)\mathrm{d}w \tag{3.5}$$

When σ tends to 0, the Gaussian converges to the previous indicatrix function.
The main advantage of such a change of function is that it recognises a Minimal Conditional Exponential Family (MCEF) introduced by [Lin et al., 2019]. For recall,

Definition 3.3.1 – *Conditional Exponential Family*

Let us suppose,

1. $q(z) \coloneqq \int_{w\in\Omega} q(z,w)\mathrm{d}w = \int_{w\in\Omega} q(z \mid w)q(w)\mathrm{d}w$

Then, the joint $q(z,w)$ is said to be a conditional exponential family (CEF) if

(P.1) $q(z \mid w) \coloneqq h_z(z,w)\exp\left(\langle \phi_z(z,w), \lambda_z\rangle_z - A_z(\lambda_z, w)\right)$ *is a member of an exponential family.*

(P.2) $q(w) \coloneqq h_w(w)exp\left(\langle \phi_w(w), \lambda_w\rangle_w - A_w(\lambda_w)\right)$ *is also a member of an exponential family.*

(P.3) The set of natural parameters of $q(z \mid w)$ and $q(w)$, denoted by Ω_z and Ω_w are both opens.

Then a Minimal Conditional Exponential Family checks the following property

Properties 3.3.1 – *Minimal Conditional Exponential Family*

A conditional EF defined in (3.3.1) is said to have a minimal representation when

(H.1) $m_w(.) : \Omega_w \to \mathcal{M}_w$ *is one-to-one*

(H.2) $m_z(., \lambda_w) : \Omega_z \to \mathcal{M}_z$ *is also one-to-one* $\forall \lambda_w \in \Omega_w$

Where $m_w(.)$ and $m_z(., \lambda_w)$ respresent the mean-parameter function of the two distributions $q(w)$ and $q(z \mid w)$ respectively.

Then, [Lin et al., 2019] showed that doing the classic hypothesis of Exponential Families (EF) to obtain VON still exist with MCEF. Therefore, as we are going to see in the next sections, it can derive a new set of updates almost as simple as the ones of VON with Gaussians (cf (2.17))

Remark 3.3.1 – *Gaussians are not Mandatory*

As for VON, the choice of the two Gaussians is not mandatory for the two sub-probabilities in (3.5). Any MCEF will allow obtaining nice and efficient VON updates.

3.4 A New Variational Inference Formulation

For obtaining nice VON updates, we need to adapt our Variational Inference problems for MCEF. Therefore, following the procedure of [Lin et al., 2019], we are going to minimise the following the Kullback-Lieber divergence over the joint distribution this time

$$\underset{q_\lambda\in\mathcal{F}}{\operatorname{argmin}}\ KL\left(q_\lambda(f_X, w) \,||\, p(f_X, w \mid D)\right) \tag{3.6}$$

Replacing (2.12) by (3.6) allows us to obtain nice derivatives for the natural gradient. In addition, this choice is also more meaningful as it works with two random variables f_X and w (cf (3.5)). In fact, as we are interested by a full-Bayesian setting, considering the posterior over the two random variables

is also more logic.

Finally, if it assumes that the data points of our training set are *iid* it obtains the following equivalent formulation

$$\underset{q_\lambda \in \mathcal{F}}{\operatorname{argmin}} \sum_{i=1}^{N} \mathbb{E}_{q_\lambda(f,w)} \left[\ln p(D_i \mid f_{x_i}, w)\right] - KL\left(q_\lambda(f_X, w) \,||\, p(f_X, w)\right) \tag{3.7}$$

With

- Tractable family $\mathcal{F}$ the set of probabilities

$$\begin{aligned} q_\lambda(f_X, w) &:= q_{\lambda_f}(f_X \mid w)\, q_{\lambda_w}(w) \\ &:= \mathcal{N}\left(f_X \mid f(w, X), \sigma^2 I\right) \mathcal{N}\left(w \mid \mu, \Sigma\right) \end{aligned}$$

- Prior $p(f_X, w)$

$$\begin{aligned} p(f_X, w) &:= p(f_X \mid w)\, p(w) \\ &:= \mathcal{N}\left(f_X \mid 0, K(X, X)\right) \mathcal{N}\left(w \mid 0, I/\delta^2\right) \end{aligned}$$

  With $K(X, X)$ a kernel.

3.5 VON with MCEF

Now the problem is posed, it can derive its VON updates. First, as in [Lin et al., 2019], it needs to define the different parameters involved.

3.5.1 Natural parameters, sufficient statistics and mean parameters

First, let write (3.5) as a minimal conditional exponential family

- It is well know that q_{λ_w} is a minimal exponential family with:

$$\begin{array}{c|c} \Phi(w) & \lambda_w \\ \hline \begin{bmatrix} \Sigma^{-1}\mu \\ \frac{-1}{2}\Sigma^{-1} \end{bmatrix} & \begin{bmatrix} w \\ ww^T \end{bmatrix} \end{array} \tag{3.8}$$

  From which it deduces the mean parameter

$$\begin{cases} m_w^{(1)} = \mu \\ M_w^{(2)} = \Sigma + \mu\mu^T \end{cases} \tag{3.9}$$

- For $q_{\lambda_f}(f_X \mid w)$ it has

$$\begin{aligned} q_{\lambda_f}(f_X \mid w) &= \exp\left(-\frac{1}{2\sigma^2} f_X^T f_X + \frac{1}{\sigma^2} f_X^T f(w, X) - \frac{1}{2\sigma^2} f(w, X)^T f(w, X) - \frac{N}{2}\ln 2\pi - \frac{N}{2}\ln\sigma^2\right) \\ &= \exp\left(Tr\left(-\frac{1}{2\sigma^2} I\left(f_X f_X^T - 2 f_X f(w, X)^T\right)\right)\right. \end{aligned}$$

$$
\begin{aligned}
&\left.\underbrace{-\frac{1}{2\sigma^2} f(w,X) f(w,X)^T - \frac{N}{2}\ln 2\pi - \frac{N}{2}\ln\sigma^2}_{:=-\mathcal{A}(\lambda_f, w)}\right) \\
&= \exp\left(\langle \lambda_f, \Phi(f_X, w)\rangle - \mathcal{A}(\lambda_f, w)\right)
\end{aligned} \tag{3.10}
$$

That can be summarised by

$$
\begin{array}{c|c}
\Phi(f_X, w) & \lambda_f \\
\hline
\left[f_X f_X^T - 2 f_X f(w,X)^T\right] & \left[-\frac{1}{2\sigma^2} I\right]
\end{array} \tag{3.11}
$$

And it deduces the mean parameter

$$
\begin{aligned}
M_f &= \mathbb{E}_{q_{\lambda_f}(f_X|w) q_{\lambda_w}(w)}\left[f_X f_X^T - 2 f_X f(w,X)^T\right] \\
&= \mathbb{E}_{q_{\lambda_w}(w)}\left[\sigma^2 I + f(w,X) f(w,X)^T - 2 f(w,X) f(w,X)^T\right] \\
&= \sigma^2 I - \mathbb{E}_{q_{\lambda_w}(w)}\left[f(w,X) f(w,X)^T\right] \\
&= -\frac{1}{2}\lambda_f^{-1} - \mathbb{E}_{q_{\lambda_w}(w)}\left[f(w,X) f(w,X)^T\right]
\end{aligned} \tag{3.12}
$$

3.5.2 Minimality

Second, for applying VON to our distribution (3.5), it needs to check its minimality.

For doing so, it can remark that $q_{\lambda_w}(w)$ is a classic gaussian under its minimal form. Therefore, $m_w(.)$ is one-to-one. Then, it just needs to prove that $m_f(., \lambda_w)$ is also one-to-one according to [Lin et al., 2019]. In particular, it can remark that:

$$
\begin{aligned}
\nabla_{\lambda_f} m_f(., \lambda_w) &= \frac{1}{2}\lambda_f^{-2} \\
&= 2\sigma^4 I \\
&> 0, \qquad \forall \lambda_w \in \Omega_w
\end{aligned}
$$

Therefore, $m_f(., \lambda_w)$ is also one-to-one and (3.5) is a minimal conditional exponential family.

3.5.3 Variational Inference with MCEF

Now, it checked the minimality of our conditional exponential family, it can derive the Natural Gradient Descent (NGD) updates following [Lin et al., 2019].
First, let write again our minimisation problem

$$
\begin{aligned}
\mathcal{L}_{VI}(\lambda) &= \sum_{i=1}^{N} \mathbb{E}_{q_\lambda(f,w)}\left[\ln p(D_i \mid f_{x_i}, w)\right] - KL\left(q_\lambda(f,w) \,||\, p(f_X, w)\right) \\
&= \sum_{i=1}^{N} \mathbb{E}_{q_{\lambda_w}(w)}\left[\mathbb{E}_{q_{\lambda_f}(f_{x_i}|w)}\left[\ln p(D_i \mid f_{x_i})\right]\right] + \\
&\qquad \mathbb{E}_{q_\lambda(f_X,w)}\left[\ln \frac{\mathcal{N}\left(f_X \mid 0, K(X,X)\right)}{\mathcal{N}\left(f_X \mid f(w,X), \sigma^2 I\right)} + \ln \frac{\mathcal{N}\left(w \mid 0, \delta^{-2} I\right)}{\mathcal{N}\left(w \mid \mu, \Sigma\right)}\right]
\end{aligned}
$$

With λ the couple (λ_f, λ_w).

Let us denote $l(f_X, w) = -\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{q_{\lambda_f}(f_{x_i}|w)}\left[\ln p(D_i \mid f_{x_i})\right]$ and $l(f_X) = -\frac{1}{N}\sum_{i=1}^{N} \ln p(D_i \mid f_{x_i})$. Thanks to this notation, it has

$$\begin{aligned}\mathcal{L}_{VI}(\lambda) &= \mathbb{E}_{q_{\lambda_w}(w)}\left[-Nl(f_X, w)\right] + \mathbb{E}_{q_\lambda(f_X,w)}\left[\ln \frac{\mathcal{N}\left(f_X \mid 0, K(X,X)\right)}{\mathcal{N}\left(f_X \mid f(w,X), \sigma^2 I\right)} + \ln \frac{\mathcal{N}\left(w \mid 0, \delta^{-2} I\right)}{\mathcal{N}\left(w \mid \mu, \Sigma\right)}\right] \\ &= \mathbb{E}_{q_{\lambda_w}(w)}\left[-Nl(f_X, w)\right] - KL\left(q_{\lambda_w}(w) \,||\, p(w)\right) \\ &\quad - \mathbb{E}_{q_{\lambda_w}(w)}\left[KL\left(\mathcal{N}\left(f_X \mid f(w,X), \sigma^2 I\right) \,||\, \mathcal{N}\left(f_X \mid 0, K(X,X)\right)\right)\right]\end{aligned}$$

It recognises the almost same Variational Inference problem but with a new loss and a new regulariser. Now we well defined the Variational Inference problem, it can now compute the natural gradient.

3.5.4 Natural Gradient

As explained before, using MCEF leads to simple updates because as for classic exponential families, it has the same result of convexity for MCEF as proved by [Lin et al., 2019]

$$\left[F_{fw}(\lambda)\right]^{-1} \nabla_\lambda \mathcal{L} = \nabla_{m_{fw}} \mathcal{L} \tag{3.13}$$

With m_{fw} the mean parameter corresponding to the vector (m_w, m_f).

Then, it can apply the chain rule to obtain

$$\begin{cases} \nabla_{m_w^{(1)}} \mathcal{L}(\lambda) = \nabla_\mu \mathcal{L}(\lambda) - 2\nabla_\Sigma \mathcal{L}(\lambda)\mu \\ \nabla_{M_w^{(2)}} \mathcal{L}(\lambda) = \nabla_\Sigma \mathcal{L}(\lambda) \\ \nabla_{M_f} \mathcal{L}(\lambda) = \nabla_{\sigma^2} \mathcal{L}(\lambda) I \end{cases} \tag{3.14}$$

And then using in the same way than for VON the Bonnet and Price theorems (cf theorems (2.7.1) and (2.7.2)), it obtains

$$\begin{cases} \nabla_\mu \mathbb{E}_{q_{\lambda_w}(w)}\left[l(f_X, w)\right] = \mathbb{E}_{q_{\lambda_w}(w)}\left[\nabla_w l(f_X, w)\right] := \mathbb{E}_{q_{\lambda_w}(w)}\left[\boldsymbol{g}_w(f_X, w)\right] \\ \nabla_\Sigma \mathbb{E}_{q_{\lambda_w}(w)}\left[l(f_X, w)\right] = \frac{1}{2}\mathbb{E}_{q_{\lambda_w}(w)}\left[\nabla^2_{ww} l(f_X, w)\right] := \frac{1}{2}\mathbb{E}_{q_{\lambda_w}(w)}\left[\boldsymbol{H}_w(f_X, w)\right] \\ \nabla_{\sigma^2} \mathbb{E}_{q_{\lambda_f}(f_{x_i}|w) q_{\lambda_w}(w)}\left[l(f_X)\right] = \frac{1}{2}\mathbb{E}_{q_{\lambda_f}(f_{x_i}|w) q_{\lambda_w}(w)}\left[\nabla^2_{f_X f_X} l(f_X)\right] := \frac{1}{2}\mathbb{E}_{q_{\lambda_f}(f_{x_i}|w) q_{\lambda_w}(w)}\left[\boldsymbol{H}_f(f_X)\right] \end{cases} \tag{3.15}$$

For finally using the fact that the *Kullback-Lieber* divergence between two Gaussians has a close form

$$\mathcal{D}_{KL}(\mathcal{N}(\mu_0, \Sigma_0)||\mathcal{N}(\mu_1, \Sigma_1)) = \frac{1}{2}\left(Tr\left(\Sigma_1^{-1}\Sigma_0\right) + (\mu_1 - \mu_0)^T \Sigma_1^{-1}(\mu_1 - \mu_0) - d + \ln\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right)\right)$$

to derive the updates for VON with our double Gaussians MCEF (cf (3.5))

$$\begin{cases} \nabla_\mu \mathcal{L}(\lambda) := \sum_{i=1}^{N} \nabla_\mu \mathbb{E}_{q_{\lambda_w}(w)}\left[\mathbb{E}_{q_{\lambda_f}(f_{x_i}|w)}\left[\log p(D_i \mid f_{x_i})\right]\right] - \delta^2 \mu - \mathbb{E}_{q_{\lambda_w}}\left[\nabla_w f(w,X)^T K(X,X)^{-1} f(w,X)\right] \\ \nabla_\Sigma \mathcal{L}(\lambda) := \sum_{i=1}^{N} \nabla_\Sigma \mathbb{E}_{q_{\lambda_w}(w)}\left[\mathbb{E}_{q_{\lambda_f}(f_{x_i}|w)}\left[\log p(D_i \mid f_{x_i})\right]\right] - \frac{1}{2}(\delta^2 I - \Sigma^{-1}) \\ \qquad\qquad - \frac{1}{2}\mathbb{E}_{q_{\lambda_w}}\left[\nabla_w f(w,X)^T K(X,X)^{-1} \nabla_w f(w,X) + \nabla^2_w f(w,X)^T K(X,X)^{-1} f(w,X)\right] \\ \nabla_{\sigma^2} \mathcal{L}(\lambda) := \sum_{i=1}^{N} \nabla_{\sigma^2} \mathbb{E}_{q_{\lambda_w}(w)}\left[\mathbb{E}_{q_{\lambda_f}(f_{x_i}|w)}\left[\log p(D_i \mid f_{x_i})\right]\right] - \left(\frac{1}{2} Tr(K(X,X)^{-1}) - \frac{N}{2\sigma^2}\right) \end{cases}$$

Using these results and following the procedure of [Khan et al., 2018], it obtains

$$\begin{cases} \mu_{t+1} = \mu_t - \beta_t \Sigma_{t+1} \left[N \boldsymbol{g}_w(f_{X,t}, w_t) + \delta^2 \mu_t + \nabla_w f(w_t, X)^T K(X,X)^{-1} f(w_t, X) \right] \\ \Sigma_{t+1}^{-1} = (1-\beta_t)\Sigma_t^{-1} + \beta_t \left[N \boldsymbol{H}_w(f_{X,t}, w_t) + \delta^2 I + \nabla_w f(w_t, X)^T K(X,X)^{-1} \nabla_w f(w_t, X) \right. \\ \qquad\qquad \left. + \nabla_w^2 f(w_t, X)^T K(X,X)^{-1} f(w_t, X) \right] \\ \dfrac{1}{\sigma_{t+1}^2} = \dfrac{1}{\sigma_t^2}(1 - \beta_t N) + \beta_t \left[N \boldsymbol{H}_f(f_{X,t}) + Tr(K(X,X)^{-1}) \right] \end{cases} \tag{3.16}$$

Where, it uses one MC sample to approximate the two expectations.

From these updates, it can see the impact of the regularisation with a GP *prior* on μ and Σ.

However, applying directly these updates could be impossible in practice. Indeed, if it works with a big data set, the inversion of the matrix $K(X,X)$ will become prohibitive. Therefore, we need to find a new approximation able to deal with such a matrix.

The inversion of the matrix $K(X,X)$ is a classic problem of Gaussian Processes. Therefore, in the next chapter, we are going to study the classic solutions used in the GP community from this review propose a new formulation of the problem of Variational Inference in infinite-dimension such that it could efficiently approximate the previous updates.

Chapter 4

Project II - VON in Infinite-Dimension

In this chapter, we are going to present a new formulation of the problem of Variational Inference in the infinite-dimension case. This new formulation is a promising solution for solving our previous problem and having a scalable algorithm for doing Gaussian Processes Regression.

4.1 Gaussian Processes and the Problem of the Inversion of the Kernel

When it works with Gaussian Processes a classic problem is to solve a Gaussian Regression problem. In term of Bayes rule, it means that all the probability are Gaussians, both the *likelihood* and the *prior*. This formulation is called Gaussian Process Regression because it is equivalent to solve a least-square regression with a gaussian *prior* on the weights. In such a situation, the *posterior* is also Gaussian because of the conjugacy between the *likelihood* and the *prior* and it has the following closed-form

$$p(f_{test} \mid X_{test}, \underbrace{y_{train}, X_{train}}_{:=D}) = \mathcal{N}(m, S) \tag{4.1}$$

With

- $m = K(X_{test}, X_{train}) \left(K(X_{train}, X_{train}) + \sigma^2 I\right)^{-1} y_{train}$
- $S = K(X_{test}, X_{test}) - K(X_{test}, X_{train}) \left(K(X_{train}, X_{train}) + \sigma^2 I\right)^{-1} K(X_{train}, X_{test})$

However, and as in our previous problem, it needs to invert the kernel matrix $K(X, X)$ which becomes prohibitive in case of a big data set.

4.2 Inducing Points

To face the problem of the inversion of the kernel matrix, the community imagined different solutions. Today, the most popular one is maybe the technique of the inducing points. The idea is to use a limited number of points instead of using the whole data-set. In other words, the aim is to summarise the data by a small number of points called *inducing points*.

To create/to compute these *inducing points*, several methods have been developed. However, the most efficient and popular is maybe the algorithm created by cf [Titsias, 2009]. The idea of [Titsias, 2009] is to find the inducing points by minimising a VI problem.

Let us consider,

- $\mathcal{X}$ the definition set of our points $(x_i)_{i=1}^N$
- $Z := [z_1, z_2, \cdots, z_M]^T \in \mathcal{X}^M$ our set of inducing points

- $f_Z := [f(z_1), f(z_2), \cdots, f(z_M)]^T$ their function value

Then, it tries to approximate the exact posterior $p(f_X, f_Z \mid y, X, Z)$ with the following approximate posterior $q(f_X, f_Z) = q_\lambda(f_Z)p(f_X \mid f_Z, y, X, Z)$ by doing VI

$$\underset{\lambda, Z \in \mathcal{X}^M}{argmin}\ KL\left(q_\lambda(f_X, f_Z) \,||\, p(f_X, f_Z \mid y, X, Z)\right) \tag{4.2}$$

In the case of GP Regression, the optimal choice for $q(f_Z)$ is a Gaussian and its covariance matrix can be inverted in $\mathcal{O}(M^3)$. Then for the prediction, it will just use $q(f_Z)$ instead of using the whole data set. Therefore, it replaces the exact posterior by an approximate posterior.
However, *inducing points* suffer from some drawbacks. In particular, if it works with a complicated task, having a lot of *inducing points* could be necessary for having a good precision. Therefore, once again, we face the same problem of inverting a large matrix.

4.3 Variational Inference in the Infinite-Dimension Case

Instead of working with inducing points, few recent papers explore the idea to use the function-space for overcoming the issue of the inversion of the matrix. In particular, they propose to put probabilities not on the weights, neither the outputs of functions like in the previous section for a given set of points (cf (3.3)) but on functions itself. Therefore the problem (2.12) becomes

$$\underset{q_\lambda}{argmin}\ KL\left(q_\lambda(f) \,||\, p(f \mid D)\right) \tag{4.3}$$

And [Sun et al., 2019] proved the following equality

$$\underset{q_\lambda}{argmin}\ KL\left(q_\lambda(f) \,||\, p(f \mid D)\right) = \sup_{n \in N, X \in \mathcal{X}^n} KL\left(q_\lambda(f(X)) \,||\, p(f(X) \mid D)\right) \tag{4.4}$$

Thanks to this property, [Sun et al., 2019] propose to do classic gradient descent on (4.3) but they take a lower bound of the gradient of KL term by sampling some points of $\mathcal{X}$ at each iteration of the optimisation procedure. Therefore, the algorithm is going to explore several points along with the optimisation procedure and at each iteration, the sampling will be most likely different so it does not limit ourself to a subset of *inducing points*.

From these results, we would like to derive a VON equivalent of the algorithm developed by [Sun et al., 2019]. In fact, one of the previous students under the supervision of M. KHAN get the same idea and published a paper in May 2019 with a new algorithm to solve VI with probabilities over functions that was inspired by VON. However, his algorithm is not as intuitive as VON. Therefore in the next sections, we are going to describe the method used by [Shi et al., 2019] and show that it corresponds in fact to do a classic VON algorithm. Along the way, we will explore the theoretical difficulties to work with infinite-dimension and therefore, many of our following calculus corresponds to draft but we are still unable to give any theoretical guarantee on their validity.

4.4 Method of [Shi et al., 2019]

4.4.1 General problem

As [Sun et al., 2019], [Shi et al., 2019] is interested in the formulation of the VI problem in the function space. In particular, he is interested in the following minimisation problem

$$\underset{q_\lambda \in \mathcal{F}}{argmin}\ KL\left(q_\lambda(f) \,||\, p(f \mid D)\right) \tag{4.5}$$

With

- $\mathcal{F}$: a family of tractable distributions.
- λ: It assumes that the distribution can be parametrised by λ. In the following, it will correspond to the natural parameter of an exponential family.
- D: a training data-set.

4.4.2 Description of the method

In this section, we describe the optimisation algorithm used by [Shi et al., 2019].

First, [Shi et al., 2019] is interested by the following problem

$$\underset{q(f)}{argmin}\ \mathcal{L} := \underset{q(f)}{argmin}\ KL\left(q(f)\ ||\ p(D\mid f)\ p(f)\right) \tag{4.6}$$

It can remark, that they do not do any assumption on $q(f)$ for now. For minimising it, they proceed as follow:

1. As a first step, before doing any assumption on $q(f)$, [Shi et al., 2019] performs a mirror descent step on $q(f)$ (for some recall on mirror descent, please refers to D.1.1). In particular, it has a closed-form solution for this mirror descent which correspond to a Bayesian filter. For example at the step $t+1$, it has

$$q_{t+1}(f) \propto p(y_n \mid f)^{N\beta_t} p(f)^{\beta_t} q_t(f)^{1-\beta_t} \tag{4.7}$$

   However, as f is equivalent to an infinite-dimensional vector, it is difficult to work with it.

2. [Shi et al., 2019] proposed to approximate the equation (4.7) by boostrapping an inference network at each iteration. They denote by $q_\gamma(f)$ the inference network, with γ the parameters it wants to estimate. In practice, they assume that the inference network acts like a GP: for any finite set of points X_M, the evaluated points f_M follow a Gaussian distribution: $q_\gamma(f_M) := \mathcal{N}(\mu_M, \Sigma_M)$ where μ_M, Σ_M depend on γ and X_M. An immediate solution for bootstrapping is to replace $q_t(f)$ by the current posterior approximation $q_{\gamma_t}(f)$

$$\hat{q}_{t+1}(f) \propto p(y_n \mid f)^{N\beta_t} p(f)^{\beta_t} q_{\gamma_t}(f)^{1-\beta_t} \tag{4.8}$$

3. An attractive property of equation (4.8) is that, given inputs X_M, all the quantities in the right-hand side follow a Gaussian distribution. Therefore $\hat{q}_{t+1}$ has a Gaussian distribution. However, it is not clear how to derive γ_{t+1} from this posterior $\hat{q}_{t+1}(f)$. In fact, if it uses an inference network, it could have a non-linear relationship between μ_M, Σ_M and γ and then deriving γ_{t+1} becomes difficult.

4. To deal with this difficulty, [Shi et al., 2019] proposed to match the marginal between $q_\gamma(f)$ and $\hat{q}_{t+1}(f)$ to find the update γ_{t+1} as in [Sun et al., 2019]. In particular, they use some random inputs X_M and then match the marginals by minimising $KL\left(q_\gamma(f_M)\ ||\ \hat{q}_{t+1}(f_M)\right)$. For the update γ_{t+1}, they take one gradient step according to γ. So, they finally obtain the following update rule for γ

$$\gamma_{t+1} = \gamma_t - \eta_t \nabla_\gamma KL\left(q_\gamma(f_M)\ ||\ \hat{q}_{t+1}(f_M)\right)|_{\gamma=\gamma_t} \tag{4.9}$$

4.4.3 Summary of the method

In summary, [Shi et al., 2019] proposes to approximate the dynamic of the exact mirror descent. It approximates it thanks to an inference network. In particular, each step of this dynamic is approximated by

$$q_{t+1}(f) \propto p(y_n \mid f)^{N\beta_t} p(f)^{\beta_t} q_t(f)^{1-\beta_t} \approx p(y_n \mid f)^{N\beta_t} p(f)^{\beta_t} q_{\gamma_t}(f)^{1-\beta_t} \propto \hat{q}_{t+1} \approx q_{\gamma_{t+1}}(f) \tag{4.10}$$

In fact, [Shi et al., 2019] explains it quite well in different parts of his paper:

- *" In this paper, we propose an algorithm to scalably train the network by tracking an adaptive Bayesian filter defined in the function space. The filter is obtained by using a stochastic, functional mirror-descent algorithm"*
- *"Our method can also be interpreted in a teacher-student framework, where the teacher can be obtained from the student network by taking a stochastic mirror descent step"*

By doing so, they want to avoid to sample points as in [Sun et al., 2019] and therefore to be sure to minimise the true problem (4.5).

However, even if this idea of tracking a learning procedure is interesting, it is not clear if the exact dynamic (cf equation (4.7)) is stable. In other words, [Shi et al., 2019] does not provide any guarantees that the approximates $q_{\gamma_t}(f)$ are going to converge towards $q_t(f)$. Above all, we are not sure if the final iterate $q_{\gamma_{t_f}}$ truly minimises the initial problem

$$q_{\gamma_{t_f}} \in \underset{q(f)}{argmin}\, KL\left(q(f) \,||\, p(D \mid f)\, p(f)\right) \; ? \quad \text{or even} \quad \gamma_{t_f} \in \underset{\gamma}{argmin}\, KL\left(q_\gamma(f) \,||\, p(D \mid f)\, p(f)\right) \; ?$$

We could, therefore, work to derive possible guarantees over the approximation of the previous dynamic and check the convergence. However it seems really difficult and in fact, in the next section, we are going to see that it is possible to derive a direct optimisation of the general problem (4.5) which is equivalent to the method of [Shi et al., 2019]. In the next section, we will give some recall on the difficulty to work in infinite-dimension and in a next next section, we will give a draft for a "functional VON".

4.5 Natural Gradient in Infinite-Dimension

In this section, we will give some recall on the infinite-dimension case. In particular, our goal is to explain how to define a probability over real functions.

4.5.1 Difficulty to work with functions

One interesting interpretation of real functions is to see them as infinite-dimensional vectors. For example, for functions defined on $\mathbb{R}$, f can be seen as an infinite-dimensional vector indexed by $\mathbb{R}$ where $\forall x \in \mathbb{R}$, $f_x = f(x)$ (for analogy with a finite vector u and its i^th components u_i).

However, because it works in infinite dimension, it does not work longer with vectors and matrices but with functions (= infinite-dimensional vector) and linear operators. In particular, one of the main difficulty of the infinite-dimension case is the lack of an equivalent Lebesgue's measure. Therefore, if it writes integrals such that

$$\int q(f) \ln \frac{q(f)}{p(f)} df \tag{4.11}$$

It is not clear which measure it uses without any precision.

Therefore, I cannot guarantee that all the equations written in the proof of [Shi et al., 2019] are correct. In addition, the lack of Lebesgue measure allows only to talk about probability measures and not probability densities. In fact, the measure theory will play a key role in writing rigorous equations.
Because of this technical difficulty, in the next section, we will just write down the first draft for deriving a "functional" VON algorithm but we cannot guarantee the correctness of the equations by lack of knowledge on measure theory.

4.5.2 Structure over space of functions: RKHS

As explained, working with functions can be interpreted as working with infinite-dimensional vectors. However, it is not as usual as working with classical finite vectors. Doing calculus with finite real vectors is easy because they belong to a vector space. Therefore, the addition of two vectors is still a real vector, the multiplication of a vector by a scalar is still a real vector, etc. We would like to have such a structure over our space of functions. Fortunately, this structure can be easily derived for functions by introducing Hilbert space and RKHS.

For recall,

Definition 4.5.1 – *Hilbert Space*

Let suppose $\mathcal{H}$ is a set,

(P.1) $\mathcal{H}$ is a vector space

(P.2) $\mathcal{H}$ is endowed with a scalar-product $\langle ., . \rangle_{\mathcal{H}}$

(P.3) $\mathcal{H}$ is complete for the norm associated with the scalar-product

Then, $\mathcal{H}$ is an Hilbert space

Definition 4.5.2 – *Reproducing Kernel Hilbert Space (RKHS)*

Let suppose $\mathcal{H}$ is a set and K is a kernel,

(P.1) $\mathcal{H}$ is an Hilbert space over functions f (defined over a set $\mathcal{X}$)

(P.2) $\forall x \in \mathcal{X}, K_x = K(x, .) \in \mathcal{H}$

(P.3) Reproducing property: $\forall x \in \mathcal{X}, f(x) = \langle f, K_x \rangle_{\mathcal{H}}$

Then, $\mathcal{H}$ is an RKHS

Therefore, if it works with functions that belong to an RKHS, it will be easy to add two functions, to multiply a function by a scalar or even "extract the $x^t h$ component" of the infinite vector f thanks to the reproducing property.

Therefore, we are looking to define an RKHS over functions.

4.5.3 Probability over functions: Example with Gaussian measures

For recall, our aim is to define a probability over function. In particular, it would like to define a kind of exponential family over functions. In this section, we will give the example of a Gaussian measure over these functions.

According to [Holmes and Sengupta, 2013], for a fixed RKHS $\mathcal{H}$, it can define a Gaussian measure ν on a Banach space B which possesses the RKHS $\mathcal{H} \subset B$. In particular, there is a mean functional $\mu \in \mathcal{H}$ and a bounded positive semi-definite linear operator $\Sigma : \mathcal{H} \to \mathcal{H}$, such that

$$\forall h \in \mathcal{H}, \qquad \int_{\mathcal{H}} \langle f, h \rangle_{\mathcal{H}} \ \nu(df) = \langle \mu, h \rangle_{\mathcal{H}}$$

$$\forall h, g \in \mathcal{H}, \qquad \int_{\mathcal{H}} \langle f - \mu, h \rangle_{\mathcal{H}} \langle f - \mu, g \rangle_{\mathcal{H}} \ \nu(df) = \langle \Sigma(h), g \rangle_{\mathcal{H}}$$

The triple $(B, \nu, \mathcal{H})$ is known as an abstract Wiener space [Eldredge, 2016], in which $\mathcal{H}$ is also called the Cameron-Martin space.

Interesting fact: it can do a link with Gaussian processes. Let consider the Gaussian process $GP(m, S)$ and let assume that the m is a function of the RKHS $\mathcal{H}$ and S is a bilinear function on the same RKHS associated with a Kernel K. If it denotes by K_{X_M}, the vector $(K_{x_1}, \cdots, K_{x_M})^T, \forall X_M \in \mathcal{X}^M$, according to [Holmes and Sengupta, 2013], they can be associated with a function μ and a linear operator Σ such that

$$m(X_M) = \left(\langle \mu, K_{x_i} \rangle_{\mathcal{H}} \right)_{i=1}^{M}$$
$$S(X_M, X_M) = \left(\left\langle \Sigma(K_{x_i}), K_{x_j} \right\rangle_{\mathcal{H}} \right)_{i,j=1}^{M}$$

4.5.4 Assumptions

As μ and Σ are functions, it is difficult to optimise according to them directly. Instead, it will assume in the following that both μ and Σ only depend on a finite set of parameters γ. For example, if it assumes the NTK kernel (cf [Jacot et al., 2018], [Khan et al., 2019]) for the Gaussian Process, γ will be equal to $\gamma = (w_s, S, m)$.

Also, it will assume that the previous Gaussian measure can be express as an exponential family and all the derivations in the next section are correct.

Remark 4.5.1 – *Does it exist exponential families in infinite-dimension?*

> *I need to work more to better understanding measures and how to well defined a kind of exponential family on Gaussian measures.*

Thanks to these results, it can derive a direct optimisation algorithm to minimise (4.5).

4.6 The draft towards VON in Infinite-Dimension

In this section, we will assume that all the operations are legal by lack of knowledge about measure theory.

4.6.1 The first draft towards VON in infinite-dimension

We are interested by the minimisation of (4.5).

Instead of approximating the dynamic, we propose to directly optimise the initial problem. Let assume that it can parametrised our gaussian measure by its "natural parameter" $\lambda_w = \left(\Sigma_\gamma^{-1}(\mu_\gamma), -\frac{1}{2}\Sigma_\gamma^{-1}\right)$. Here it adds the dependence according to γ.

Thanks to this new parametrisation, it can apply a classic gradient descent on γ

$$\gamma_{t+1} = \gamma_t - \beta_t \nabla_\gamma KL\left(q_\lambda(f) \,||\, p(f \mid D)\right) \tag{4.12}$$

Let us also assume a Gaussian measure prior over f with "natural parameter" λ_0 and let denote by m^f the expectation parameter associated with $q_\lambda(f)$. In particular, m^f will also depends on γ. Therefore, it adds this dependency and denotes it by m^f_γ. Thanks to these assumptions, it obtains by chain rule

$$\begin{aligned}
\gamma_{t+1} &= \gamma_t - \beta_t \left\langle \nabla_\gamma m^f_\gamma, \nabla_{m^f_\gamma} KL\left(q_\lambda(f) \,||\, p(f \mid D)\right) \right\rangle_{\mathcal{H}} \\
&= \gamma_t + \beta_t \left\langle \nabla_\gamma m^f_\gamma, \nabla_{m^f_\gamma} \mathbb{E}_{q_{\lambda_\gamma}}\left[\ln p(D \mid f)\right] - \nabla_{m^f_\gamma} KL\left(q_\lambda(f) \,||\, p(f)\right) \right\rangle_{\mathcal{H}}
\end{aligned}$$

$$= \gamma_t - \beta_t \left\langle \nabla_\gamma m_\gamma^f, \lambda_{\gamma,t} \right\rangle_{\mathcal{H}} + \beta_t \left\langle \nabla_\gamma m_\gamma^f, \lambda_0 \right\rangle_{\mathcal{H}} + \beta_t \left\langle \nabla_\gamma m_\gamma^f, \nabla_{m_\gamma^f} \mathbb{E}_{q_{\lambda_\gamma}} \left[\ln p(D \mid f) \right] \right\rangle_{\mathcal{H}}$$

Finally, if it assumes a Bonnet and Price theorem version for Gaussian measure, it will obtain updates really similar to the ones of VON.

4.6.2 Comparison with [Shi et al., 2019]

To understand how the algorithm proposed by [Shi et al., 2019] is similar to the one of the previous section, it can derive the same kind of equation for the method used by [Shi et al., 2019]. In particular, if it summarises the optimisation process in one step, it obtains

$$\begin{aligned} & \gamma_{t+1} = \gamma_t - \beta_t \nabla_\gamma KL \left(q_{\gamma_{t+1}}(f(X_M)) \mid\mid \hat{q}_{t+1}(f(X_M)) \right) |_{\gamma=\gamma_t} \\ \iff \quad & \gamma_{t+1} = \gamma_t - \beta_t \nabla_\gamma KL \left(q_{\gamma_{t+1}}(f(X_M)) \mid\mid p(y_n \mid f(X_M))^{N\beta_t} p(f(X_M))^{\beta_t} q_{\gamma_t}(f(X_M))^{1-\beta_t} \right) |_{\gamma=\gamma_t} \end{aligned}$$

If it proceeds in different steps, it has

1. Let's rewrite the *Kullback-Lieber* and use $\lambda = \lambda(\gamma)$ for simplicity

$$\begin{aligned} & \mathrm{KL} \left(q_{\lambda^M}(f_M) \;\; \mid\mid \;\; p(y_n \mid f_M)^{N\beta_t} \;\; p(f_M)^{\beta_t} \;\; q_{\lambda_t^M}(f_M)^{1-\beta_t} \right) \\ & \quad = \mathbb{E}_{q_{\lambda^M}(f_M)} \left[\ln q_{\lambda^M}(f_M) - N\beta_t \ln p(y_n \mid f_M) - \beta_t \ln p(f_M) - (1-\beta_t) \ln q_{\lambda_t^M}(f_M) \right] \end{aligned} \tag{4.13}$$

2. As we are working with GP, all the probabilities considered are Gaussian. In particular, the likelihood is a Gaussian which is conjugate with the prior

$$\ln p(y_n \mid f_M) = \lambda_{lik}^M(y_n)^T \Phi_{\mathcal{G}}(f_n) - \mathcal{A}_{\mathcal{G}}(\lambda_{lik}^M(y_n)) \tag{4.14}$$

   With

   - $\Phi_{\mathcal{G}}$ the classic sufficient statistics of the multivariate-Gaussian but extended to infinite-dimensional case
   - $\mathcal{A}_{\mathcal{G}}$ is the classic log-partition function extended to the infinite-dimensional case

3. Also the prior is a Gaussian process with natural parameter λ_0

$$\ln p(f_M) = (\lambda_0^M)^T \Phi_{\mathcal{G}}(f_M) - \mathcal{A}_{\mathcal{G}}(\lambda_0^M) \tag{4.15}$$

4. In the same way, it has

$$\ln q_\lambda(f_M) = (\lambda^M)^T \Phi_{\mathcal{G}}(f_M) - \mathcal{A}_{\mathcal{G}}(\lambda^M) \tag{4.16}$$

$$\ln q_{\lambda_t}(f_M) = (\lambda_t^M)^T \Phi_{\mathcal{G}}(f_M) - \mathcal{A}_{\mathcal{G}}(\lambda_t^M) \tag{4.17}$$

5. Then, as for our algorithm, it could apply the chain rule with $m_\gamma^{f_M}$. By forgetting the constant terms, it obtains

$$\begin{aligned} & \nabla_\gamma \left(m_\gamma^{f_M} \right)^T \nabla_{m_\gamma^{f_M}} \mathrm{KL} \left(q_{\lambda^M}(f_M) \;\; \mid\mid \;\; p(y_n \mid f_M)^{N\beta_t} p(f_M)^{\beta_t} q_{\lambda_t^M}(f_M)^{1-\beta_t} \right) |_{\gamma=\gamma_t} \\ & = \nabla_\gamma \left(m_\gamma^{f_M} \right)^T \nabla_{m_\gamma^{f_M}} \mathbb{E}_{q_{\lambda^M}(f(X_M))} \left[\left(\lambda^M - \beta_t \lambda_0^M - (1-\beta_t)\lambda_t^M - N\beta_t \lambda_{lik}^M(y_n) \right)^T \Phi_{\mathcal{G}}(f_M) - \mathcal{A}_{\mathcal{G}}(\lambda^M) \right] |_{\gamma_t} \\ & = \nabla_\gamma \left(m_\gamma^{f_M} \right)^T \nabla_{m_\gamma^{f_M}} \left[\left(\lambda^M - \beta_t \lambda_0^M - (1-\beta_t)\lambda_t^M - N\beta_t \lambda_{lik}^M(y_n) \right)^T m_{\lambda^M}^{f_M} - \mathcal{A}_{\mathcal{G}}(\lambda^M) \right] |_{\gamma=\gamma_t} \\ & = \nabla_\gamma \left(m_\gamma^{f_M} \right)^T \left(\beta_t \lambda_t^M - \beta_t \lambda_0^M - N\beta_t \lambda_{lik}^M(y_n) \right) \end{aligned}$$

6. Finally, let define $\langle f, g \rangle_{\mathcal{H}} := \mathbb{E}_{x \sim p(x)} \left[f(x)^T g(x) \right]$ and $p(x)$ is a uniform distribution over $\mathcal{X}$. Then it can see the previous equation as a MC approximation of this expectation. Therefore, by putting everything together

$$\gamma_{t+1} \approx \gamma_t - \beta_t \left\langle \nabla_\gamma m_\gamma^f, \lambda_t \right\rangle_{\mathcal{H}} + \beta_t \left\langle \nabla_\gamma m_\gamma^f, \lambda_0 \right\rangle_{\mathcal{H}} + N \beta_t \left\langle \nabla_\gamma m_\gamma^f, \lambda_{lik}(y_n) \right\rangle_{\mathcal{H}} \tag{4.18}$$

   Therefore, it falls back on the equation of our algorithm. The only difference here is they use a "doubly stochastic" approach for estimating the expectation for the likelihood $\lambda_{lik}(y_n)$.

Here, because of the infinite-dimension, we cannot guarantee the exactness of our calculus.

I obtained this result one month before the end of my internship. The fact that the algorithm I wanted to develop was in fact similar to the one developed by [Shi et al., 2019] combined with the fact of the very strong theoretical difficulties to prove properly these results pushed me to look for another project where I hoped I could obtain quicker results before the end of my internship.

At the same moment, my supervisor gets interested in a new project: understanding the fascinating relationship between VON with Gaussians and the Taylor expansion.

Chapter 5

Project III - VON as a Taylor Expansion

At the opposite of the two previous chapters, in this one, we are going to focus on the classic VON algorithm. In particular, we are going to consider again the weight-space and we are going to show how applying VON with Gaussians corresponds in fact to a Taylor Expansion of order 2. Finally, we will explore how it could use another probability to obtain a Taylor expansion of a higher order.

5.1 Some Hypothesis

In the weight-space, doing Variational Inference corresponds to the minimisation of

$$\min_{q_\lambda \in \mathcal{P}} KL\left(q_\lambda(w) \mid\mid p(w \mid D)\right) \tag{5.1}$$

As in the first chapter, we will consider that all the probabilities belong to an exponential family and in particular, for sake of simplicity, that the *prior* $p(w)$ and the approximate posterior $q_\lambda(w)$ belongs to the same family. Under such a hypothesis, the VI problem becomes

$$\max_\lambda \mathcal{L}_{VI}(\lambda) \tag{5.2}$$
$$= \max_\lambda \ \mathbb{E}_{q_\lambda}\left[\left(\left\langle \lambda_{lik}(w), \Phi_{lik}(D)\right\rangle_{lik} - A_{lik}(\lambda_{lik}(w))\right) - \left(\left\langle \lambda - \lambda_{prior}, \Phi(w)\right\rangle - A(\lambda) + A(\lambda_{prior})\right)\right]$$

Finally, let us suppose for sake of simplicity that w is a one-dimensional variable ($w \in \mathbb{R}$) and let us assume that the likelihood is not necessarily conjugate but can be written as

$$\ln p(D \mid w) = \left\langle \lambda_{lik}(D), \Phi_{lik}(w)\right\rangle_{lik} + Const \tag{5.3}$$

For example, in the case of the training of a BNN for a regression task with a least-squares loss, it has:

- $\lambda_{lik} = \big(Y, I\big)^T \in \mathbb{R}^N \times \mathbb{R}^{N\times N}$ with N the number of data points.
- $\Phi_{lik}(w) = \big(f(w,X), f(w,X)f(w,X)^T\big)^T \in \mathbb{R}^N \times \mathbb{R}^{N\times N}$ with f the neural architecture.
- $\left\langle \, , \right\rangle_{lik} = \left\langle ., . \right\rangle_{\mathbb{R}^N} + Tr(. \times .)_{\mathbb{R}^{N\times N}}$ is the sum of the classic euclidean scalar-product for the vectors and the Trace for the matrices.

The main purpose of this assumption is to obtain an equation similar to the one obtained in case of conjugacy (cf Appendix B) and therefore to justify the use of the Natural Gradient. Therefore, (5.2) becomes

$$\max_\lambda \ \mathbb{E}_{q_\lambda}\left[\left(\left\langle \lambda_{lik}(D), \Phi_{lik}(w)\right\rangle_{lik}\right) - \left(\left\langle \lambda - \lambda_{prior}, \Phi(w)\right\rangle - A(\lambda)\right)\right] \tag{5.4}$$

Where it removed the constant terms

5.2 VON with Gaussians

Then, as explained in the previous sections, doing VON with an approximate posterior Gaussian is particularly appealing because of the Bonnet's and Price's theorems (cf (2.7.1) and (2.7.2)). Indeed, these theorems allow passing the derivatives inside the expectation. Above all, for our aim to derive a Taylor expansion, these two theorems are fascinating because they do a direct link between the derivatives according to μ and Σ with the derivatives of the loss. therefore, in the next subsections, we are going to write again all the step of the VON algorithm to understand what are the key properties of the Gaussian distribution that allow making appear the Taylor Expansion.

5.2.1 First step of VON: Natural Gradient Step

First, let us assume that both the *prior* and the *posterior* are Gaussian distributions and let us denote by λ_{prior} and λ their natural parameter respectively. Then, following the calculus of [Khan et al., 2018], we consider the classic natural gradient update

$$\lambda_{t+1} = (1-\beta_t)\lambda_t + \beta_t\lambda_{prior} + \beta_t \nabla_m \left[\mathbb{E}_{q_\lambda}\left[\ell(w)\right]\right]_t \tag{5.5}$$

With

- $\ell(w)$ the classic log-likelihood, $\ell(w) = \ln p(D \mid w)$
- m the expectation parameter of p_λ. In the case of Gaussians, it has

$$m = \left(\mu, \sigma^2 + \mu^2\right)^T = \left(\mathbb{E}_{q_\lambda}\left[w\right], \mathbb{E}_{q_\lambda}\left[w^2\right]\right)^T$$

Now, we consider the scalar product between λ_{t+1} and $\Phi(w)$,

$$\left\langle \lambda_{t+1}, \Phi(w)\right\rangle = \sum_{n=1}^{2} \lambda_{t+1,n} w^n \tag{5.6}$$

With

- $\lambda_{t+1,n}$ the n^{th} component of λ_{t+1}
- $\Phi(w)$ the classic sufficient statistic of a Gaussian distribution: $\Phi(w) = \begin{pmatrix} w \\ w^2 \end{pmatrix}$

Therefore, by applying this scalar product (5.6) to (5.5), it obtains:

$$\begin{aligned}
\left\langle \lambda_{t+1}, \Phi(w)\right\rangle &= (1-\beta_t)\left\langle \lambda_t, \Phi(w)\right\rangle + \beta_t \left\langle \lambda_{prior}, \Phi(w)\right\rangle + \beta_t \left\langle \nabla_m \left[\mathbb{E}_{q_\lambda}\left[\ell(w)\right]\right]_t, \Phi(w)\right\rangle \\
&= (1-\beta_t)\left\langle \lambda_t, \Phi(w)\right\rangle + \beta_t \left\langle \lambda_{prior}, \Phi(w)\right\rangle + \beta_t \sum_{n=1}^{2} \frac{\partial}{\partial m_n}\left[\mathbb{E}_{q_\lambda}\left[\ell(w)\right]\right]_t w^n
\end{aligned} \tag{5.7}$$

Finally, by using the decomposition of the loss $\ell(w)$ (5.3), it obtains

$$\begin{aligned}
\sum_{n=1}^{2} \frac{\partial}{\partial m_n}\left[\mathbb{E}_{q_\lambda}\left[\ell(w)\right]\right] w^n &= \sum_{n=1}^{2} \frac{\partial}{\partial m_n}\left[\mathbb{E}_{q_\lambda}\left[\left\langle \lambda_{lik}(D), \Phi_{lik}(w)\right\rangle_{lik}\right]\right]_t w^n \\
&\quad \text{by linearity of the scalar product } \left\langle \, , \right\rangle_{lik} \\
&= \sum_{n=1}^{2} \frac{\partial}{\partial m_n}\left[\left\langle \lambda_{lik}(D), \mathbb{E}_{q_\lambda}\left[\Phi_{lik}(w)\right]\right\rangle_{lik}\right]_t w^n \\
&= \left\langle \lambda_{lik}(D), \sum_{n=1}^{2} \frac{\partial}{\partial m_n}\left[\mathbb{E}_{q_\lambda}\left[\Phi_{lik}(w)\right]\right]_t w^n \right\rangle_{lik}
\end{aligned} \tag{5.8}$$

Where it removed the constant terms.

Putting everything together, it obtains:

$$\begin{aligned}\langle \lambda_{t+1}, \Phi(w)\rangle = (1-\beta_t)\langle \lambda_t, \Phi(w)\rangle + \beta_t \langle \lambda_{prior}, \Phi(w)\rangle \\ + \beta_t \left\langle \lambda_{lik}(D), \sum_{n=1}^{2} \frac{\partial}{\partial m_n}\left[\mathbb{E}_{q_\lambda}\left[\Phi_{lik}(w)\right]\right]_t w^n \right\rangle_{lik}\end{aligned} \tag{5.9}$$

From this first step, it can see the importance of the form of the sufficient statistics of the Gaussian to make appear the Taylor expansion.

5.2.2 Second step of VON: Bonnet's and Price's Theorems

When it works with Gaussian distributions for the *posterior*, it could further simplify the previous equation (5.9) thanks to the Bonnet and Price's Theorem ((2.7.1) and (2.7.2)). Indeed, for a Gaussian distribution, it has:

$$\begin{cases} m_1 = \mu \\ m_2 = \sigma^2 + \mu^2 \end{cases} \iff \begin{cases} \mu = m_1 \\ \sigma^2 = m_2 - m_1^2 \end{cases} \tag{5.10}$$

Therefore it can apply the chain rule for computing the gradient according to m:

$$\begin{cases} \dfrac{\partial}{\partial m_1}\left[\mathbb{E}_{q_\lambda}\left[\Phi_{lik}(w)\right]\right] = \dfrac{\partial}{\partial \mu}\left[\mathbb{E}_{q_\lambda}\left[\Phi_{lik}(w)\right]\right] - 2\dfrac{\partial}{\partial \sigma^2}\left[\mathbb{E}_{q_\lambda}\left[\Phi_{lik}(w)\right]\right]\mu \\ \dfrac{\partial}{\partial m_2}\left[\mathbb{E}_{q_\lambda}\left[\Phi_{lik}(w)\right]\right] = \dfrac{\partial}{\partial \sigma^2}\left[\mathbb{E}_{q_\lambda}\left[\Phi_{lik}(w)\right]\right] \end{cases} \tag{5.11}$$

Finally, by applying the Bonnet and Price's theorems ((2.7.1) and (2.7.2)), it obtains:

$$\begin{cases} \dfrac{\partial}{\partial m_1}\left[\mathbb{E}_{q_\lambda}\left[\Phi_{lik}(w)\right]\right] = \mathbb{E}_{q_\lambda}\left[\nabla_w \Phi_{lik}(w)\right] - \mathbb{E}_{q_\lambda}\left[\nabla_w^2 \Phi_{lik}(w)\right]\mu \\ \dfrac{\partial}{\partial m_2}\left[\mathbb{E}_{q_\lambda}\left[\Phi_{lik}(w)\right]\right] = \dfrac{1}{2}\mathbb{E}_{q_\lambda}\left[\nabla_w^2 \Phi_{lik}(w)\right] \end{cases} \tag{5.12}$$

By plugging these results to the equation (5.9), it obtains:

$$\begin{aligned}\langle \lambda_{t+1}, \Phi(w)\rangle = (1-\beta_t)\langle \lambda_t, \Phi(w)\rangle + \beta_t \langle \lambda_{prior}, \Phi(w)\rangle \\ + \beta_t \left\langle \lambda_{lik}(D), \left[\mathbb{E}_{q_\lambda}\left[\nabla_w \Phi_{lik}(w)\right]\right] w + \frac{1}{2}\left[\mathbb{E}_{q_\lambda}\left[\nabla_w^2 \Phi_{lik}(w)\right]\right](w^2 - 2w\mu_t) \right\rangle_{lik}\end{aligned} \tag{5.13}$$

As we can see, we are getting closer to the Taylor expansion, but not yet. For obtaining a true Taylor expansion, it needs to apply the last VON step: the MC sampling. From this second step, we can also see the importance that the Gaussian checks the Bonnet's and Price's theorems for the passing of the derivatives under the expectation.

5.2.3 Third step of VON: MC Sampling

The last step of VON consists of doing MC sampling for estimating the previous expectations. Let us apply it to our previous equation (5.13) for S MC samples of $q_{\lambda_t}(w)$:

$$\begin{aligned}\langle \lambda_{t+1}, \Phi(w)\rangle = (1-\beta_t)\langle \lambda_t, \Phi(w)\rangle + \beta_t \langle \lambda_{prior}, \Phi(w)\rangle \\ + \beta_t \left\langle \lambda_{lik}(D), \frac{1}{S}\sum_{s=1}^{S}\left(\nabla_w \Phi_{lik}(w_s) w + \frac{1}{2}\nabla_w^2 \Phi_{lik}(w_s)(w^2 - 2w\mu_t)\right) \right\rangle_{lik}\end{aligned} \tag{5.14}$$

Which is also equal to

$$\langle \lambda_{t+1}, \Phi(w)\rangle = (1-\beta_t)\langle \lambda_t, \Phi(w)\rangle + \beta_t \langle \lambda_{prior}, \Phi(w)\rangle + \frac{\beta_t}{S}\sum_{s=1}^{S}\left(\left\langle \lambda_{lik}(D), \nabla_w \Phi_{lik}(w_s)w + \frac{1}{2}\nabla_w^2 \Phi_{lik}(w_s)(w^2 - 2w\mu_t)\right\rangle_{lik}\right) \quad (5.15)$$

For obtaining the Taylor expansion desired, it needs to do two more steps. First, let us add the constant $C = \frac{\beta_t}{S}\sum_{s=1}^{S}\left(\langle \lambda_{lik}(D), \Phi_{lik}(w_s) - \nabla_w \Phi_{lik}(w_s)\mu_t + \frac{1}{2}\nabla_{ww}^2 \Phi_{lik}(w_s)\mu_t^2\rangle_{lik}\right)$ in both sides of the previous equation to obtain:

$$\langle \lambda_{t+1}, \Phi(w)\rangle + C = (1-\beta_t)\langle \lambda_t, \Phi(w)\rangle + \beta_t \langle \lambda_{prior}, \Phi(w)\rangle + \frac{\beta_t}{S}\sum_{s=1}^{S}\left(\left\langle \lambda_{lik}(D), \Phi_{lik}(w_s) + \nabla_w \Phi_{lik}(w_s)(w-\mu_t) + \frac{1}{2}\nabla_w^2 \Phi_{lik}(w_s)(w-\mu_t)^2\right\rangle_{lik}\right) \quad (5.16)$$

And finally, let us do a simple reparametrisation. Let us introduce the new random variables w'_s defined such that $\forall s \in [\![1, S]\!], \quad w'_s - w_s = w - \mu_t$. By doing so, it obtains the Taylor expansion of order 2 so desired

$$\langle \lambda_{t+1}, \Phi(w)\rangle + C = (1-\beta_t)\langle \lambda_t, \Phi(w)\rangle + \beta_t \langle \lambda_{prior}, \Phi(w)\rangle + \frac{\beta_t}{S}\sum_{s=1}^{S}\left(\left\langle \lambda_{lik}(D), \underbrace{\Phi_{lik}(w_s) + \nabla_w \Phi_{lik}(w_s)(w'_s - w_s) + \frac{1}{2}\nabla_w^2 \Phi_{lik}(w_s)(w'_s - w_s)^2}_{\text{Taylor expansion of order 2 of } \Phi_{lik} \text{ in } w_s}\right\rangle_{lik}\right) \quad (5.17)$$

It finally obtained the Taylor expansion so desired.

5.2.4 Interpretations and first remarks

Interpretations

It can give an interesting interpretation of the previous equation (5.17) by rewriting the previous equation with probabilities. In particular, let us compose by the exponential and let us divide by the normalising constants of each probability defined by its scalar-product. It obtains:

$$q_{\lambda_{t+1}}(w) \propto q_{\lambda_t}(w)^{1-\beta_t}\left(p(w)\prod_{s=1}^{S}\left(p_{Taylor_{2,w_s}}(w'_s)\right)^{1/S}\right)^{\beta_t} \quad (5.18)$$

With

- q_{λ_t} the approximate posterior at time t.
- $q_{\lambda_{t+1}}$ the new approximate.
- $p(w)$ the *prior*
- $\forall s \in [\![1, S]\!], \quad p_{Taylor_{2,w_s}}(w'_s)$ is the probability defined as the exponential family with sufficient statistics equal to the Taylor development at the order 2 in w_s of the sufficient statistics of the likelihood $\Phi_{lik}(w)$.

Therefore, the posterior $q_{\lambda_{t+1}}$ corresponds to the classic Bayesian filtering of VON. In other words, it is a weighted average between two probabilities:

1. $p_1(w) = q_{\lambda_t}(w)$ with weight $1-\beta_t$ which is the previous iterate or approximated posterior

2. $p_2(w) = p(w) \prod_{s=1}^{S} \left(p_{Taylor_2,s}(w'_s)\right)^{1/S}$ with weight β_t

Here, the probability p_2 has an interesting interpretation. $p(w)$ corresponds to the probability of the *prior* and therefore the a priori knowledge. Above all, $\prod_{s=1}^{S} \left(p_{Taylor_k,s}(w'_s)\right)^{1/S}$ represents an average of the value of the true likelihood in different regions. In fact, as $w'_s = w - \mu_t + w_s$ and w is a random variable following $q_{\lambda_{t+1}}$, if μ_{t+1} is close to μ_t, then w'_s is a random variable following a Gaussian distribution with a mean equal to w_s. Therefore the Taylor expansions in each w_s will be valid as w'_s will be close to w_s it obtains an reliable average of the value of the likelihood thanks to these Taylor expansions.

First Remarks

1. In the previous sections, it follows the approximation made by VON to derive a Taylor expansion. However, it could still obtain a Taylor expansion by doing another kind of approximations. In particular, instead of doing MC sampling it could approximate $\mathbb{E}_{q_\lambda}\left[\nabla_w \Phi_{lik}(w)\right]$ by $\nabla_w \Phi_{lik}(\mu_t)$ and $\mathbb{E}_{q_\lambda}\left[\nabla^2_w \Phi_{lik}(w)\right]$ by $\nabla^2_w \Phi_{lik}(\mu_t)$. However, it would not need to reparametrise our problem but we would rely on a unique estimate of the likelihood around μ_t.
2. Here, it derived the equations under the assumption that the *likelihood* could be written as a probability over w and not D (cf (5.3)). We made this choice for its similarity with the case of conjugacy (cf appendix B) and where the natural gradient reaches the optimum in one step. Nevertheless, if it is not the case, the previous equations will still be valid but with a Taylor approximation on $\ell(w)$.
3. Finally, we did the assumption that both the *prior* and the *posterior* belongs to the same exponential family. In fact, his requirement is not necessary. Indeed, in the opposite case, we will obtain that the *posterior* corresponds to a Taylor expansion on both the *prior* and the *likelihood*.

5.3 Which Distribution for a Taylor Expansion of Higher-Order?

5.3.1 Requirements

In the previous section, we derived a beautiful result for VON (cf equation (5.17)) only when the approximate posterior is Gaussian. In fact, the reason the Taylor expansion appears specifically for the Gaussian relies on two main reasons:

1. If it goes back to the calculus of the section 5.2.1, it can see that the Taylor expansion of order 2 appears because the Gaussian distributions have a precise sufficient statistics of the form $\Phi(w) = \begin{pmatrix} w \\ w^2 \end{pmatrix}$
2. The Taylor expansion also appears because, for the Gaussian distribution, it exists a relationship between the derivative according to the mean parameter and the derivatives of the loss function. These relationships correspond to the theorems of Bonnet and Price.

Therefore, if it wants to find a distribution that allows deriving a Taylor expansion to a higher order, it needs to take care of these two considerations. In particular, we begin by introducing the polynomial exponential family which is a direct consequence of this first consideration.

5.3.2 Polynomial Exponential Families

As the Gaussian probability is only parametrised by its first two moments and the sufficient statistics of the Gaussian is $\phi(w) = \left(w, w^2\right)^T$, we consider the exponential families $q_{k,\lambda}$ defined by the following sufficient statistics

$$\Phi_k(w) = \left(w, w^2, \dots, w^k\right)^T$$

In particular, for $k = 2$, it falls on the Gaussian family.

Therefore, the expectation parameter for these families is simply the vector of the moments:

$$\begin{aligned}\mathbb{E}_{q_k}\left[\Phi_k(w)\right] &= \left(\mathbb{E}_{q_{k,\lambda}}\left[w\right], \ldots, \mathbb{E}_{q_{k,\lambda}}\left[w^k\right]\right)^T \\ &= \left(m_1, \ldots, m_k\right)^T\end{aligned}$$

We call this family the "Polynomial Exponential Family".

5.3.3 First step of VON: Natural gradient with Polynomial Exponential Family

As the polynomial exponential family is really similar to the Gaussian distribution, it could derive similar calculus than the ones of the section 5.2.1.

First, as before, let us assume that both the *prior* and the *posterior* are members of the same polynomial exponential family and let us denote by λ_{prior} and λ their natural parameter respectively. Then, following the calculus of [Khan et al., 2018], we consider the classic natural gradient update:

$$\lambda_{t+1} = (1 - \beta_t)\lambda_t + \beta_t\lambda_{prior} + \beta_t\nabla_m\left[\mathbb{E}_{q_{k,\lambda}}\left[\ell(w)\right]\right]_t \tag{5.19}$$

With $\ell(w)$ is the classic log-likelihood $\ell(w) = \ln p(D \mid w)$ and m the expectation parameter of $p_{k,\lambda}$: $m = \left(m_1, \ldots, m_k\right)^T = \left(\mathbb{E}_{q_{k,\lambda}}\left[w\right], \ldots, \mathbb{E}_{q_{k,\lambda}}\left[w^k\right]\right)^T$.

Then by doing the same calculus than the ones of the section 5.2.1, it obtains

$$\langle\, \lambda_{t+1}, \Phi(w)\rangle = (1-\beta_t)\,\langle\, \lambda_t, \Phi(w)\rangle + \beta_t\,\langle\, \lambda_{prior}, \Phi(w)\rangle + \beta_t\left\langle \lambda_{lik}(D), \sum_{n=1}^{k}\frac{\partial}{\partial m_n}\left[\mathbb{E}_{q_\lambda}\left[\Phi_{lik}(w)\right]\right]_t w^n \right\rangle_{lik} \tag{5.20}$$

5.3.4 Second step of VON: Difficulties for a generalised Bonnet's and Price's theorems

Then if it wants to derive the same kind of equations than the ones for the Gaussian case, it needs a Generalised Bonnet and Price theorems that generalise to another kind of distribution.

First, let us look to the distribution belonging to the polynomial exponential family of order 4 for example. Among all the polynomial EF of order 4, we are looking for some of them that could check a kind of generalised Bonnet and Price theorem. It is at that point that the difficulties appear.

In fact, for the Gaussian case, the Bonnet and Price theorem exists because of a simple Integration by Parts. Indeed, when we derive the probability of a Gaussian according to x, we obtain $-(x - \mu)/sigma * p(x)$. In other words, a polynomial of order 1 multiplied by $p(x)$: $P_1(x) * p(x)$. Therefore, if it has a polynomial of any order multiply by p(x), it can recursively apply integration by parts and therefore obtain the Bonnet and Price theorems.

Problem: when it considers our polynomial EF distribution of order 4; when it derives two times according to x, p(x), it obtains $P_6(x) * p(x)$ where $P_6(x)$ denotes a polynomial of order 6. Then the problem appears! Indeed, we are looking to link the derivatives of p(x) according to its expectation parameters to the different derivatives of p(x) according to x. Therefore, as explained before, we are looking for a parameter able to make appear a polynomial of order 6... Without any further calculus, this is impossible.

Therefore, as it seems to me unlikely to find a distribution belonging to the polynomial exponential family and verifying a kind of generalised Bonnet and Price theorem, we were wondering if going with a polynomial with infinite order (e.g. exponential) could be the answer. For example, it exists a distribution called Gompertz that match this description. I worked quite intensively the past few weeks of my internship to find if it was possible to obtain any results. However, now we faced a new difficulty: we must only consider curved EF for a practical reason. (In the opposite case, it would mean that it needs to update an infinite number of parameters). Then, the relationship between the natural parameter and the expectation parameter becomes harder to determine also the equations become much heavier.

By lack of time, we get unable to continue further. Nevertheless, we would like to continue in the future this project to determine the feasibility of such an approach.

Chapter 6

Conclusion

For my final internship, I get the amazing chance to join one of the most prestigious research institute of Japan: Riken AIP. As a member of the team of M. KHAN specialised on Approximate Bayesian Inference, I had the wonderful opportunity to learn and work on incredible topics such as Bayesian Neural Networks. In particular, the aim is to improve the estimation of the uncertainty on the predictions of deep neural networks and developing new tools of interpretability.

The team published recently a fascinating algorithm called VON for Variational Online Newton in [Khan et al., 2018]. This new kind of algorithm possesses all the qualities such as the efficiency and the scalability for working in high-dimension and big data set. Therefore, fascinated by such an algorithm, I dedicated my time to develop and extend the use of this algorithm to new contexts.

In particular, I firstly focus to design a new version of the algorithm able to work with meaningful prior. For doing so, the main idea is to do Variational Inference in the function-space and use Minimal Conditional Exponential Families [Khan et al., 2019]. Using meaningful prior represents a promising tool for better regularisation of deep neural networks and better interpretability of their predictions. This work is presented in Chapter III.

However, one of the main drawbacks of the method developed in Chapter III is its lack of scalability. Therefore, in Chapter IV, I reviewed the literature of Gaussian Processes and in particular their last idea of the infinite-dimension to solve the previous issue. However, because of the strong theoretical difficulties to work in infinite-dimension, I cannot guarantee all the calculus done in this Chapter.

Because of the lack of time and the difficulty to work in infinite-dimension, I dedicated the rest of my time to understand better the current VON algorithm in weight-space. In particular, I show that using VON with a Gaussian approximate posterior corresponds to do a Taylor Expansion of order 2 on the loss. Then, I worked to create new distributions able to produce a Taylor Expansion of higher-order but I faced a new kind of difficulty. This work is available in Chapter V.

Finally, even if I did not support all these projects until their end because of lack of time or resources, I really would like to continue on some of the projects presented and hopefully published a paper by the end of 2019.

Bibliography

[Eldredge, 2016] Eldredge, N. (2016). Analysis and Probability on Infinite-Dimensional Spaces.

[Flam-Shepherd et al., 2017] Flam-Shepherd, D., Requeima, J., and Duvenaud, D. (2017). Mapping Gaussian Process Priors to Bayesian Neural Networks. *Bayesian Deep Learning Workshop, Neural Information Processing Systems (NeurIPS)*.

[Hernández-Lobato and Adams, 2015] Hernández-Lobato, J. M. and Adams, R. P. (2015). Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. *International Conference on Machine Learning*, 32.

[Holmes and Sengupta, 2013] Holmes, I. and Sengupta, A. (2013). The Gaussian Radon Transform and Machine Learning. *Infinite Dimensional Analysis, Quantum Probability and Related Topics*, 18:1–26.

[Jacot et al., 2018] Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *32nd Conference on Neural Information Processing Systems (NIPS 2018)*, (5).

[Khan et al., 2019] Khan, M. E., Immer, A., Abedi, E., and Korzepa, M. (2019). Approximate Inference Turns Deep Networks into Gaussian Processes. *arXiv preprint arXiv:1906.01930*.

[Khan et al., 2018] Khan, M. E., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. (2018). Fast and Scalable Bayesian Deep Learning by Weight-Perturbation in Adam. *Proceedings of the 35 th International Conference on Machine Learning*.

[Lin et al., 2019] Lin, W., Khan, M. E., and Schmidt, M. (2019). Fast and Simple Natural-Gradient Variational Inference with Mixture of Exponential-family Approximations. *Proceedings of the 36 th International Conference on Machine Learning*.

[Raskutti and Mukherjee, 2015] Raskutti, G. and Mukherjee, S. (2015). The information geometry of mirror descent. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9389:359–368.

[Shi et al., 2019] Shi, J., Khan, M. E., and Zhu, J. (2019). Scalable Training of Inference Networks for Gaussian-Process Models. *Proceedings of the 36 th International Conference on Machine Learning*.

[Shun-ichi Amari, 1998] Shun-ichi Amari (1998). Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(251-276):1–4.

[Sun et al., 2019] Sun, S., Zhang, G., Shi, J., and Grosse, R. (2019). Functional Variational Bayesian Neural Networks. *ICLR 2019*, pages 1–22.

[Titsias, 2009] Titsias, M. K. (2009). Fluctuation correlation spectroscopy of near-field trapped nanoparticles. *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*.

[Williams and Rasmussen, 2006] Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*.

Appendix A

Information Theory

A.1 Shannon's Information

Before to continue, it seems to us important to do a brief recall and to give some perspectives about the information of Shannon that we are going to use intensively all along with this report. In fact, it exists many mathematical definitions of the information. However, it is the definition of Shannon that is the most widely used. He proposed it in 1948, in his famous paper "A Mathematical Theory of Communication". In it, he tried to understand the fundamental limits of signal processing and data compression. In this purpose, he developed a new notion of the information. It defines the information carried by an observation w of a random variable W defined on Ω as:

$$\mathcal{I}(p(w)) = \ln \frac{1}{p(w)} \tag{A.1}$$

This choice corresponds to careful thinking and could be explained by some intuitions.

As the title of his paper reminds us, Shannon was interested in the communication of the information more than the information itself which is by nature not fully observable. It is important because w corresponds to the received message/observation after passing through a noisy channel of communication. Therefore, when it observes a rare message w, it means that they are few possibilities for the initial inputs/state that could have produced it. For recall, a probability p is a measure on the definition space of W: Ω. Therefore, when $p(w)$ is small, it means that the portion of the space Ω that generated the observation/message w is small and so it is easier to determine which is the original inputs/state that generated w. Therefore, $\mathcal{I}(p(w))$ carry the information about the discernibly of the inputs/state in Ω that generated w.

Another view would be one of statistical physics. In fact, in statistical physics, it assumes that a macro-state or observation w like the pressure or the temperature of an object depends directly on the states of the atoms that composed it. Basically, it assumes a probability $p(w) = \frac{M(w)}{M}$ for the observation w where $M(w)$ stands for all the micro-states or configurations of the atoms (positions and velocity) that allows us to observe w and M for the total number of micro-states. Therefore, smaller is $p(w)$, smaller is $M(w)$ and therefore, it has more information about the discernible between the micro-states that generate the observation w.

Finally, the choice of the composition with $\ln$ could be understood as a better scaling of the information along the real axis. For example, let suppose that $p(w_1) = \frac{p(w_2)}{2}$. Therefore, it means, always with our analogy with statistical physics, that there is 2 times less configurations that generate w_1 than w_2. Therefore, at the level of the information, it has: $\mathcal{I}(p(w_1)) = \mathcal{I}(p(w_2)) + \ln(2)$. That is to say, if it chooses the basis 2 for our logarithm, it has that observing w_1 is equivalent to observe w_2 plus a bits of information. The $\ln$ allow to deal with information in a linear way.

To conclude, it could be interesting to remind that $\mathcal{I}$ is a measure about the discernibly of the micro-states or corresponds to the discernible information about the micro-states for avoiding confusions with another kind of information.

A.2 Shannon's Entropy

In his paper, Shanon introduces also the average of the information that he calls entropy:

$$H(p) = \mathbb{E}_p\left[\mathcal{I}(p(W))\right] \tag{A.2}$$

This choice of the name comes again from results of statistical physics. In fact, in statistical physics, it often does a classic assumption that, at equilibrium, all reachable observations correspond to exactly one micro-states and these observations are equally likely. Therefore, if it assumes the same hypothesis, it obtains for M possible observations/micro-states:

$$\begin{aligned} H(p) &= \mathbb{E}_p\left[\mathcal{I}(p(W))\right] \\ &= -\sum_{i=1}^{M} p(w_i)\ln p(w_i) \\ &= -\sum_{i=1}^{M} \frac{1}{M}\ln\frac{1}{M} \\ &= \ln(M) \end{aligned} \tag{A.3}$$

It finds again the famous law of Boltzman: $S = k\ln M$ where M corresponds to the number of configurations. Therefore, changing of the basis the logarithm near, it has the same formulation.

From this result, it can remind that the average of the information acts like entropy. If p changes for M constant and Shannon's entropy increases, it translates an increase in our ability to discernible between the micro-states. Also, this behaviour is not incoherent with the classic interpretation of the entropy in thermodynamics. In fact, in thermodynamics, if the entropy increases, it says that it lost information about the initial conditions. Even if, it could seem paradoxical at the beginning, it is coherent.

If it imagines again an object that increased of entropy, it could translate that p changes of support. In particular, in the observable world, the change of p corresponds to a change of support towards a bigger one. Therefore, even if it lost information about the very precise micro-states of the initialisation, it gained on average on the discernible of all micro-states thanks to this change of support. It can observce more micro-states so it has necessarily more information.

A.3 Kullback-Lieber divergence

Now the concept of information of Shannon is better understood, it can introduce the divergence of *Kullback-Lieber*. It calls it divergence because it looks like a distance but it is neither symmetric nor verify the triangular inequality. Therefore, the *Kullback-Lieber* divergence gives a kind of distance between two probabilities but it has to be careful with the order of its arguments because of the lack of symmetry. For recall,

Definition A.3.1 – *Kullback-Lieber Divergence*

Let us suppose that

(P.1) p and q are two probability distributions with the same support,

Then, the Kullback-Lieber *divergence is defined by*

$$KL\left(q \,||\, p\right) = \int_{\Omega} q(w) \ln \frac{q(w)}{p(w)} dw = \mathbb{E}_{q(w)}\left[\ln \frac{q(W)}{p(W)}\right]$$

A.3.1 Properties

Even if the *Kullback-Lieber* divergence is not a distance, it still has some interesting properties. Let denotes $\mathcal{U}\left(\Omega\right)$ the set of the probability measures on Ω. Therefore, it has:

Properties A.3.1 – *Kullback-Lieber Divergence*

(P.1) $\forall p, q \in \mathcal{U}\left(\Omega\right), \quad KL\left(q \,||\, p\right) \geq 0$

(P.2) $\forall p, q \in \mathcal{U}\left(\Omega\right), \quad \left(KL\left(q \,||\, p\right) = 0\right) \Longleftrightarrow p = q$ *a.s.*

Proof – *Kullback-Lieber Divergence Properties*

Let p *and* $q \in \mathcal{U}\left(\Omega\right)$*, it has:*

1. *The first two properties can be demonstrated using the Jensen Inequality D.2.1.*
   As $-\ln$ *is strictly convex it can apply the inequality of Jensen D.2.1, it obtains*

$$\begin{aligned} KL\left(q \,||\, p\right) &= \mathbb{E}_{q(w)}\left[-\ln \frac{p(W)}{q(W)}\right] \\ &\geq -\ln\left(\mathbb{E}_{q(w)}\left[\frac{p(W)}{q(W)}\right]\right) \\ &\geq -\ln\left(\int_{\Omega} p(w)\mathrm{d}w\right) \\ &\geq 0 \end{aligned}$$

2. *Again, according to the inequality of Jensen D.2.1,* $\mathbb{E}_q\left[\ln \frac{q}{p}\right] = 0$ *if and only if* $\ln \frac{q}{p}$ *is a constant almost surely. Therefore, if it denotes by* c *this constant, it has*

$$\begin{aligned} \ln \frac{q}{p} = c &\Longleftrightarrow q = p \exp c \ \text{ a.s.} \\ &\text{but } q \text{ and } p \text{ are probability measure. Their mass is equal to 1. So} \\ &\Longleftrightarrow c = 0 \text{ and } q = p \text{ a.s.} \end{aligned}$$

■

A.3.2 Kullback-Lieber Interpretations

The *Kullback-Lieber* has different interpretations

- It can rewrite the *Kullback-Lieber* divergence to reveal the information of p and q:

$$KL\left(q \,||\, p\right) = \int_{\mathcal{W}} q(w) \ln \frac{q(w)}{p(w)} dw$$

$$
\begin{aligned}
&= \mathbb{E}_q \left[\ln \frac{q}{p} \right] \\
&= \mathbb{E}_q \left[\mathcal{I}(p(W)) - \mathcal{I}(q(W)) \right]
\end{aligned}
$$

For recall, the *Kullback-Lieber* divergence is always positive. Therefore, if q is the approximate function, and it wants to do predictions with it, the *Kullback-Lieber* divergence will measure the information lost by using the approximate distribution q instead of p in average among the predictions of q

- Another view, can be obtained if it rewrites the divergence:

$$
\begin{aligned}
KL\left(q \,||\, p\right) &= \int \log \left(\frac{q(w)}{p(w)} \right) q(w) dw \\
&= \int \log \left(\frac{q(w)}{p(w)} \right) \frac{q(w)}{p(w)} \underbrace{p(w)dw}_{dp}
\end{aligned}
$$

It can recognise the relative entropy of q with respect to p for the measured space defined by p. Therefore, with the statistical physics point of view, minimising the *Kullback-Lieber* divergence leads to reduce the loss of information about the initial conditions, *i.e.* p. In other words, it tries to reduce the number of transformations that happens if it passes from p to q.

Therefore, when it does VI, it uses a meaningful order between q_λ and the *posterior* inside the *Kullback-Lieber* divergence. The other way could be used and corresponds to the *Expectation-Propagation* algorithm.

Appendix B

Motivation for the Natural Gradient for VI with Exponential Family

B.1 Conjugacy with Exponential Family

With exponential families and when the *prior* and the *posterior* belongs to the same distribution, one says that the likelihood is conjugate to the *prior* if it can be written as

$$p(D \mid w) = h(w) \exp\big(\langle \lambda_{lik}(D), \Phi(w) \rangle_{prior} - A(\lambda_{lik}(D))\big) \tag{B.1}$$

It means that the sufficient statistics and the scalar product is the same than the one of the *prior* and the *posterior*.

In the following, it will forget that λ_{lik} depends on D for simplicity.

B.2 VI with Exponential Family and Conjugacy

For simplicity, let usl denote by $a^T b = \langle a, b \rangle_{prior}$. Also, if it assumes the conjugacy of the *likelihood* with the *prior*, it obtains the following minimisation problem

$$\begin{aligned}
& \max_{\lambda} \ \mathbb{E}_{q_\lambda} \Big[\big(\big(\lambda_{lik} + \lambda_{prior} - \lambda\big)^T \Phi(w) + A(\lambda) \big) \Big] \\
\iff \quad & \max_{\lambda} \ \big(\lambda_{lik} + \lambda_{prior} - \lambda\big)^T \mathbb{E}_{q_\lambda} \left[\Phi(w)\right] + A(\lambda) \\
\iff \quad & \max_{\lambda} \ \big(\lambda_{lik} + \lambda_{prior} - \lambda\big)^T m(\lambda) + A(\lambda)
\end{aligned} \tag{B.2}$$

Where m is the expectation parameter of the *posterior* distribution.

B.3 Natural gradient for VI with Conjugate Exponential Family

In such a case of conjugacy with an exponential family, applying the natural gradient becomes very meaningful. In fact, the classic natural gradient update is:

$$\begin{aligned}
& \lambda_{t+1} = \lambda_t + \beta_t \nabla_m \big[\big(\lambda_{lik} + \lambda_{prior} - \lambda\big)^T m(\lambda) + A(\lambda) \big]_t \\
\iff \quad & \lambda_{t+1} = \lambda_t + \beta_t \big[\lambda_{lik} + \lambda_{prior} - \lambda_t - F^{-1}(\lambda_t) m(\lambda_t) + F^{-1}(\lambda_t) m(\lambda_t) \big] \\
\iff \quad & \lambda_{t+1} = \lambda_t + \beta_t \big[\lambda_{lik} + \lambda_{prior} - \lambda_t \big]
\end{aligned}$$

But, in case of conjugacy, it knows the optimal parameter for the posterior. In particular, it has:

$$\lambda = \lambda_{lik} + \lambda_{prior}$$

Therefore, by choosing $\beta_t = 1$, it obtains the optimal solution.

However, the conjugacy is restrictive and it often needs to work with no conjugate family in practice.

Appendix C

Gaussian Processes

Gaussian processes is a class of generative model which implies the extensive use of the gaussian distribution for modelising the different probabilities involves.

C.1 Regression Case

To explain Gaussian processes, the easiest is to begin with the study of the classic linear regression.

C.1.1 Weight-Space View

The simplest way to begin is to introduce the regression problem trough the view of the weigth. Basically, the problem of regression corresponds at the case where it tries to approximate the outputs $\mathbf{y} = (y_i)_{i=1...N}$ as a linear combination of the inputs $X = (\mathbf{x}_i)_{i=1...N}$

$$y_i = x_i^T w + \epsilon_i \qquad \text{with} \qquad \epsilon_i \sim \mathcal{N}(0, \sigma I) \tag{C.1}$$

However, this model is very limited. In fact, it can only modelise linear relationship. Therefore, it often uses non-linear embedding $\phi(x)$ to enhance the expressiveness of the inputs. For example, ϕ can be the polynomial embedding of degree k: $\phi(x) = \left(x_1, \cdots, x_N, \cdots, x_1^k, \cdots, x_N^d\right)^T$. The problem of regression becomes now

$$y_i = \phi(x_i)^T w + \epsilon_i \qquad \text{with} \qquad \epsilon_i \sim \mathcal{N}\left(0, \sigma^2\right) \tag{C.2}$$

Now, the model can learn much richer relationship between the inputs and outputs.

If it considers a Bayesian framework, it can add a gaussian prior on the weights w: $w \sim \mathcal{N}(0, \Sigma_p)$ independent from the noise. From this assumption, it can deduce the law of the outputs

$$y_i \mid \mathbf{x}_i, \sigma, \Sigma_p \sim \mathcal{N}\left(y_i \mid 0, \phi(\mathbf{x}_i)^T \Sigma_p \phi(\mathbf{x}_i) + \sigma^2\right) \tag{C.3}$$

It can obtain this result if it remarks that the noise are independent from the weights.

Therefore, under a vectorial formulation, it has

$$\mathbf{y} \mid X, \sigma, \Sigma_p \sim \mathcal{N}\left(\mathbf{y} \mid 0, \Phi^T \Sigma_p \Phi + \sigma^2 I\right) \tag{C.4}$$

With Φ the matrix with the vectors $\phi(\mathbf{x}_i)$ stacked.

C.1.2 Functional-Space View

As Σ_p is a covariance matrix, it is in particular symmetric and positive-definite. Therefore, it can introduce the covariance-function or kernel defined as

$$k(x, x') = \phi(x)^T \Sigma_p \phi(x') \tag{C.5}$$

Also, as Σ_p is symmetric, its eigenvalues are positive and therefore it can take its square root: $\Sigma_p^{\frac{1}{2}}$. Then, if it can remark that $\phi(x)^T\Sigma_p\phi(x')$ is in fact a classic scalar product in $\mathbb{R}^p$

$$\phi(x)^T\Sigma_p\phi(x') = \phi(x)^T\Sigma_p^{\frac{1}{2}}\Sigma_p^{\frac{1}{2}}\phi(x') \tag{C.6}$$

$$= \left(\Sigma_p^{\frac{1}{2}}\phi(x')\right)^T \Sigma_p^{\frac{1}{2}}\phi(x') \tag{C.7}$$

$$= \left\langle \Sigma_p^{\frac{1}{2}}\phi(x'), \Sigma_p^{\frac{1}{2}}\phi(x') \right\rangle_{\mathbb{R}^p} \tag{C.8}$$

Alternatively, if it denotes by f_w the function $\mathbf{x} \to \phi(x)^T w$ and $f_i = f(x_i)$, it has

$$\mathbf{y} = f + \epsilon \qquad \text{with} \qquad f \sim \mathcal{N}(0, K) \tag{C.9}$$

With $K(X, X)$ the matrix associated to the kernel $k(x, x')$.

Therefore, it falls again on the *kernel trick* where it can express any scalar product of embeddings as a kernel. The main advantages is that this embedding could be infinite but the kernel has a limited shape which corresponds to the data. It means, with the *kernel trick*, we are able to express infinite scalar product in finite shape.

Remark C.1.1 – *Sampling of Gaussian Processes*

> *A sample $f(X)$ of $\mathcal{N}(0, K(X, X))$ would be a vector of shape $N \times 1$ and it corresponds to a prediction for each input in the set X.*

C.2 Definition

Therefore, thanks to the functional-space view, it can define a gaussian process

Definition C.2.1 – *Gaussian Processes*

> *A Gaussian process is a collection of random variables, finite number of which have a joint Gaussian distribution. [Williams and Rasmussen, 2006]*

Appendix D

Mathematical Background

D.1 Optimisation

D.1.1 Mirror Descent

Any gradient descent step can be reformulated as a mirror descent step

$$\lambda_{t+1} = \lambda_t + \alpha \nabla_\lambda \mathcal{L}(\lambda_t) \iff \lambda_{t+1} = \underset{\lambda}{argmax} \quad \lambda^T \nabla_\lambda \mathcal{L}(\lambda_t) - \frac{1}{2\rho_t} \|\lambda - \lambda_t\|^2$$

With $\|.\|$ is the classic L^2 norm.

However, the main difference between a classic gradient descent step and a mirror descent step is that the choice of the geometry of our optimisation parameter is much clearer because of the appearance of the distance between the current value and the update. Here it is the L^2 distance.

In practice, all the parameters on which we are doing gradient descent does not belong to a Euclidean space. Therefore, it is important to take into account the geometry of the space on which the parameter of optimisation lives. For example, in the case of VI with exponential families, the natural parameter belongs to a Riemannian space with a metric defined by the Fischer Information matrix. Therefore, as explained before, it is more efficient to do natural gradient descent that takes into account this Riemannian metric. Equivalently, it can show that the natural gradient step is, in fact, equivalent to a mirror descent step but with a new norm/metric

$$\lambda_{t+1} = \lambda_t + \alpha F(\lambda)^{-1} \nabla_\lambda \mathcal{L}(\lambda_t) \iff \lambda_{t+1} = \underset{\lambda}{argmax} \quad \lambda^T \nabla_\lambda \mathcal{L}(\lambda_t) - \frac{1}{2\rho_t} \|\lambda - \lambda_t\|^2 \tag{D.1}$$

With $\|x - y\| = (x-y)^T F(\lambda_t)^{-1}(x-y)$ follows the metric defined by the Fisher matrix.

Therefore, the mirror descent is mostly equivalent to do a gradient descent but where it takes into account the geometry of the optimisation parameter.

In particular, the mirror descent appears to be often simpler to compute in the case of exponential families.

D.2 Probabilities

D.2.1 Jensen's inequality

The theorem of Jensen is particularly useful in probability and optimisation for its result on convexity.

Theorem D.2.1 – *Jensen's Inequality*

Let us suppose

(H.1) $(\Omega, \mathcal{A}, p)$ *a measured space with a total mass equal to 1*

(H.2) g *a p-integrable function with values in an interval* I

(H.3) φ *a convex function from* I *to* $\mathbb{R}$

Then

$$\varphi\left(\int_\Omega g \ \mathrm{d}p\right) \leq \int_{\mathbb{R}} \varphi \circ g \ \mathrm{d}p$$

The right integral could be equal to $+\infty$.
If φ *is strictly convex, there is equality if and only if* g *is constant p-almost-surely.*

For demonstrating the theorem, it could use a minoration by affine functions of a convex function.

D.3 Absolutely Continuity

In this section, it summarises the different definition of an absolute continuous function.

D.3.1 Absolutely Continuous Vector Functions

Definition D.3.1 – *Absolutely Continuous Functions (AC)*

A vector function $f : [a, b] \to \mathbb{R}^p$ *with* $[a, b]$ *a compact interval of* $\mathbb{R}$ *is* absolutely continuous *(AC) if the following properties are satisfied:*

(P.1) Its derivative $\nabla_x f(x)$ *exists almost everywhere for* $x \in [a, b]$

(P.2) The derivative is Lebesgue integrable. In other words, $\int_a^b \|\nabla_x f(x)\| \, \mathrm{d}x < +\infty$, *where* $\|.\|$ *denotes the Euclidean norm*

(P.3) The fundamental theorem of calculus holds, that is $f(x) = f(a) + \int_a^x \nabla_t f(t) \mathrm{d}t$ *for any* $x \in [a, b]$

Since, it could need to work with function where its domain is $\mathbb{R}$, we define the locally AC vectors functions.

Definition D.3.2 – *Locally Absolutely Continuous Functions (ACL)*

A vector function $f : \mathbb{R} \to \mathbb{R}^p$ *is said to be locally AC if for any compact interval of its domain* $\mathbb{R}$, f *is AC on this compact.*

Finally, it can extend the notion of ACL to multivariate functions.

Definition D.3.3 – *Multivariate-Vector ACL Functions*

Let $f(x) : \mathbb{R}^d \to \mathbb{R}^p$ *a multivariate-vector function. Given* $x_{-i} \in \mathbb{R}^{d-1}$ *fixed, let's define the function* $f_i(.) = f(., x_{-i}) : \mathbb{R} \to \mathbb{R}^p$. *We say that* f *is ACL if:*

$$\forall i \in [\![1, d]\!], \quad \forall x_{-i} \in \mathbb{R}^{d-1}, \quad f_i \text{ is ACL}$$

cf definition (D.3.2)

D.4 Vocabulary

D.4.1 Closed-Form Solution

A formula or expression is say to be closed-form if it uses only classic operations:, addition, subtraction, division or exponential. It depends of course of the field considered but it can say that is easy to compute and do not need approximation for computing it. In practice, if the formula involves an integral, it will not be considered as a closed-form solution because it does not know how to compute this integral exactly.