博士論文

Robust and Computationally-Efficient Approximate Bayesian Inference

（ロバストで計算効率の良い近似ベイズ推論に関する研究）

二見　太

Contents

Abstract

To analyze data, inferring the data generating mechanism is a widely used approach from natural science to social science. Since completely describing the true data generating mechanism is difficult except for controlled simulations or experimental environments, probabilities are widely used as models to include such uncertainty. On the other hand, in statistical inference, we do not necessarily focus on finding the true data generating mechanism. Rather, the purpose is to obtain a model with a high generalization ability that makes good predictions about the unknown data based on a limited number of data at hand.

When performing statistical inference, uncertainty appears due to the observation noise and the limited sample size. Bayesian inference is an effective method for making predictions while evaluating such uncertainty. In Bayesian inference, the probability is considered as a degree of confidence, and thus we can evaluate the uncertainty through a probabilistic model. Bayesian inference has been used in practical applications including social science and medical science recently increasingly because Bayesian inference is useful for solving inverse problems. In inverse problems, our goal is to find the probability of the cause for the given effects. With Bayesian inference, we can estimate it easily by the posterior distribution. Another advantage of Bayesian inference is that we can evaluate the uncertainty in the models and predictions. Then, the uncertainty can be used as a criterion to select models or measure the reliability of the predictions. The major disadvantage of Bayesian inference is that the posterior and predictive distributions cannot be obtained analytically and we need approximation methods in many practical models. Although Bayesian inference has a long history, its practical usage began only a few decades ago because of the necessity of huge computational resources and approximation methods. The recent success of Bayesian inference is mainly because of the advances of numerical calculators and the development of approximation techniques such as sampling or parametric methods. Thus, developing better approximation methods is essential for Bayesian inference.

In this dissertation, we discuss approximation methods for Bayesian inference focusing on outliers in the observed data. When the observed data include outliers, it means there is an abnormality in the true data generating mechanism. This is the situation where unrelated contamination is somehow added to data that we are interested in. The behavior of such contamination is completely different from the main body of the observed data. In many practical situations, the main body of the observed data represents the phenomena we want to analyze and the proportion of contamination is small. In such situations, although the proportion of outliers is small, outliers are usually located in the tails of the empirical distribution of the data and thus they have a significant effect on the results of estimation. Developing robust algorithms against such outliers is very important in actual application these days since recent advances in sensor technology give a vast amount of data with

spiky noise and crowd-annotated data is full of human errors.

In this dissertation, we present the following three contributions about robust inference and approximation methods.

The first contribution is developing a computationally efficient algorithm for long-tailed distributions. The most widely used approach for robustness in Bayesian inference is the model-based approach. For example, we replace the Gaussian distribution in a model with a long-tailed distribution, such as the Student-t distribution to enhance robustness against outliers. However, the Student-t distribution is not a member of the exponential family and does not have useful properties of the exponential family. Thus, it is difficult to develop a computationally efficient algorithm for the Student-t distribution. To address this problem, with the special algebra called q-algebra, we show that the useful properties of the exponential family can be inherited to a generalized exponential family which includes the Student-t distribution. Then, we develop a generalized expectation propagation algorithm for the generalized exponential family which provides a deterministic approximation to the posterior or predictive distributions with simple moment matching.

The second contribution is the proposal of variational inference based on robust divergences. While replacing a model to a heavy-tailed distribution is a useful approach for the robustness, it can only be applied to simple models. Exploring a robust model by such replacements is not a promising approach since we need a vast computational cost each time when we estimate complex models. Hence a systematic approach for the robustness is required. For this purpose, we develop a method by changing the inference itself in Bayesian inference instead of changing the model. Bayes' theorem plays a central role in Bayesian inference, and we interpret the theorem as a solution of an optimization problem. Based on this interpretation, we find that Bayes' theorem treats all the observed data with the same weight, and thus outliers have the same impact on the result as ordinary data. Then, we propose to use robust divergences that give small weights to outliers. Furthermore, we construct a computationally efficient algorithm based on variational inference and discuss its robustness using the influence function.

The third contribution is the development of a new approximation approach based on the Frank -Wolfe algorithm. The above two approximations for robustness are parametric approaches, that is, we approximate the true posterior distribution with a parametric distribution. Such parametric assumptions make the algorithm computationally tractable and can be applied to high-dimensional problems. On the other hand, the disadvantage is that due to the strong assumptions, such as the mean field assumption, it suffers from a large bias from the true posterior distribution which cannot be bounded theoretically in general. There is an another approximation approach, a sampling-based method. With this approach, we can approximate the true posterior distribution precisely if we use a large number of samples. The bias is bounded theoretically. The disadvantage is that vast computational resources are required to sample from multi-modal and high-dimensional distributions. Then, based on these approximation approaches, we develop a new method that combines the advantage of each approximation method, that is, the theoretical guarantee of the sampling-based approach and the computational efficiency of the parametric approach. Our new algorithm approximates the posterior distribution by an empirical distribution like sampling-based approaches. The atoms of the empirical distribution are determined through a convex optimization

problem. This optimization problem is solved efficiently with the Frank-Wolfe algorithm and the quality of the solution is assured theoretically.

Acknowledgements

First of all, I would like to express my deepest gratitude to my supervisor, Prof. Masashi Sugiyama for his constant encouragement and strong guidance in my Ph.D. research. Without his patient support, I could not complete researches and thesis. During my research, he provided not only the technical advice for my research but also questions connected to the fundamental aspects in my research and which made me consider how my research is related to other research fields. In addition to that, he gave me many valuable opportunities to get acquainted with experts outside the laboratory.

I would also express my gratitude to Prof. Sato and Prof. Honda. In particular, Prof. Sato discussed a lot about the research idea and gave me a lot of advice about writing papers. Prof. Honda frankly communicated and advised me and it refreshed my idea so much. Moreover, I am very grateful to Prof. Masato Okada and Prof. Hiromichi Nagao for reviewing and evaluating my thesis.

Next, I would like to express my great appreciation to all the members of Sugiyama-Sato-Honda laboratory. In particular, I would like to thank Ikko Yamane, Takashi Ishida, and Kento Nozawa. I was really grateful for their daily casual conversation and numerous discussions. There are too many names to write here, but I really appreciate other students in the laboratory and the secretaries who always helped me so much.

My research was financially supported by the RIKEN Center for Advanced Intelligence Project (AIP), Japan Science and Technology Agency (JST) AIP-PRISM (Grant Number JPMJCR18ZH), TOYOTA- DOWANGO scholarship, and Google fellowship. Thanks to these financial support, I could prepare the research environment, conduct the research and attend the conferences. Since I quit my job and started my Ph.D course, I had a financial concern. With these support, I could totally concentrate on the research with peace of mind.

Last but not least, I would deeply appreciate my family, in particular, my parents for their understanding and support.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Statistical inference

Suppose we have data from the phenomena which we are interested in. To analyze the data, considering the mechanism of how the data was generated is an effective and widely used way from social science to natural science in general (Hastie et al., 2001). Since describing the true data generating mechanism completely is difficult except for fully controlled simulations or experimental environments, probabilities are widely used as models for the mechanism while expressing uncertainty in statistical inference (Bishop, 2006). Many models contain some tunable variables, and we call them parameters. Under such probabilistic models, each observed data is treated as a random variable. If there are an infinite number of data, we can estimate the parameter accurately (Van der Vaart, 1998). However, since collecting an infinite number of observations is impossible in practice, we need to infer a model based on limited finite data at hand. The schematic picture of statistical inference is shown in Figure. 1.1. In this figure, "nature" expresses the phenomena what we are interested in, and it generates the observed data $\{x_i\}_{i=1}^N$ following the unknown probability $p^*(x)$. Our objective is to infer the true data generating mechanism from the finite data. We adjust the model parameter with the limited data at hand so that the model is close to the true data generating mechanism under a specified criterion. Various criteria have been proposed to select an appropriate model. The most widely used criterion is the likelihood, which is a measure of how likely the observed data is under the given model (Bishop, 2006).

In natural science, we would like to find an exact data generating mechanism or its abstraction under a given time and space scale. On the other hand, in statistical inference, we do not necessarily focus on finding the true data generating mechanism. More emphasis is put on the ability to make good predictions about unknown future data. We call this ability a generalization ability and the objective of statistical inference is to obtain the model with a high generalization ability (Bishop, 2006). In particular, when estimating the parameter from finite limited data, the use of a complex model that may be close to the true data generating mechanism does not necessarily result in a model with a high generalization ability. Rather, it often suffers from overfitting due to the limited sample size. For this reason, much research has been carried out to obtain a model with a high generalization ability, for example, by restricting the space of feasible models with regularization techniques (Bishop, 2006).

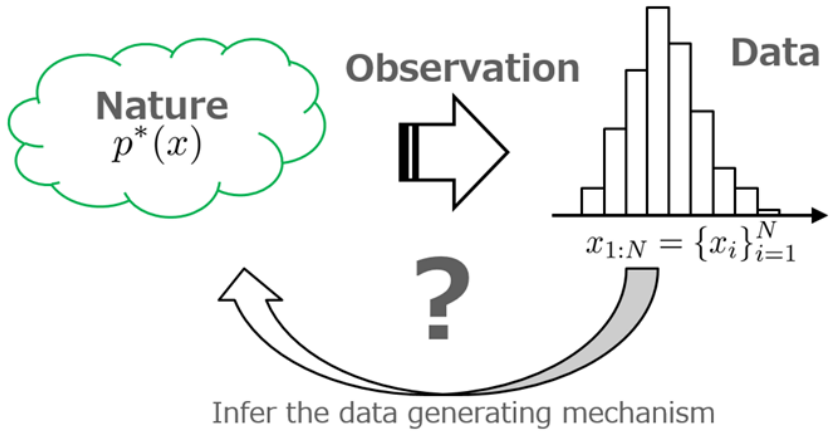

Nature
$p^*(x)$
Observation
Data
$x_{1:N} = \{x_i\}_{i=1}^N$
?
Infer the data generating mechanism

FIGURE 1.1: Schematic description of statistical inference.

1.2 Bayesian inference

In this section, we describe how to estimate the parameter and how to make the prediction. For this purpose, first, we describe Bayesian inference.

In Bayesian inference, the probability is regarded as a degree of confidence. This is very different from the frequentist probability, that is, the probability is regarded as the limit of the frequency of randomly repeated trials (Van der Vaart, 1998). Hence, in the frequentist theory, parameters in the model are treated as unknown constants. On the other hand, in Bayesian inference, based on the interpretation of the probability as a degree of confidence, we can define the probability for anything which has uncertainty (Bishop, 2006). Hence, we can define the probability of uncertainty about a model and prediction through its parameters.

In particular, in Bayesian inference, the parameter is treated as a random variable and its distribution before the observation is called a prior distribution.

If there is some prior knowledge about the parameter or the model, we incorporate them into the prior distribution as a degree of confidence. After we observed the data, the degree of confidence changes by the information of the observed data. This means that the prior distribution is modified based on the observed data. This modified distribution is called the posterior distribution. This is achieved by using Bayes' theorem (Bishop, 2006) (see Section 2.1.1 for the detail). In Bayesian inference, the parameter is treated as a random variable that follows the prior distribution and the prior distribution is updated to the posterior distribution based on the observed data with Bayes' theorem. In this way, our confidence is updated.

In Bayesian inference, the uncertainty due to the observation noise and the finite sample size is expressed through the probability distribution of the parameter. In particular, when the number of the observed data is large, the influence of the prior distribution on the posterior distribution becomes small. This means that if the number of observed data is large, inaccurate prior knowledge will not be a problem. On the other hand, when the number of the observed data is small, it may be possible to obtain a good estimate by using prior knowledge with high confidence (Bishop, 2006).

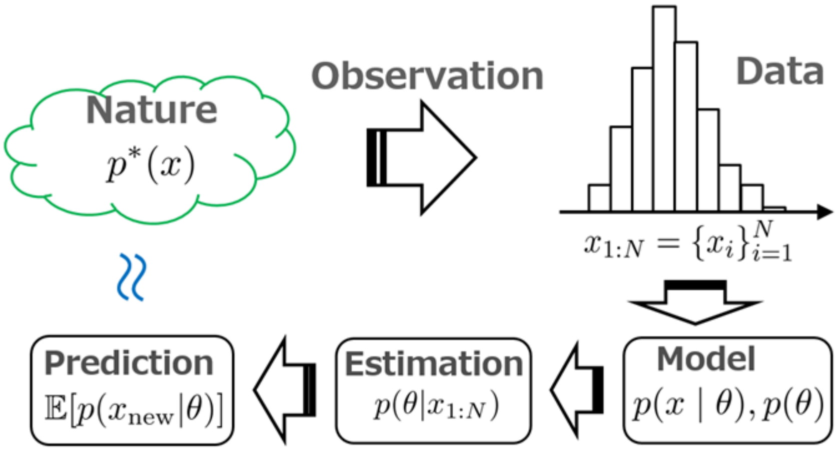

Nature
$p^*(x)$
Observation
Data
$x_{1:N} = \{x_i\}_{i=1}^N$
$\wr\wr$
Prediction
$\mathbb{E}[p(x_{\mathrm{new}}|\theta)]$
Estimation
$p(\theta|x_{1:N})$
Model
$p(x \mid \theta), p(\theta)$

FIGURE 1.2: Schematic procedure of Bayesian inference.

When prior knowledge cannot be used, a non-informative prior distribution is used to remove the influence of a prior distribution as much as possible (Bishop, 2006).

The prediction is obtained by integrating out the parameter in the model by taking the expectation with respect to the posterior distribution (Bishop, 2006). This means that we do not restrict the parameter to a point estimate like the frequentist theory and we consider all the possibilities of the parameter in the prediction through a weighted average using the posterior distribution. A schematic illustration of Bayesian inference is shown in Figure. 1.2, In the figure, θ expresses the parameter in the model. In Figure. 1.2, we assume that the model is composed of the likelihood function $p(x|\theta)$ and the prior distribution $p(\theta)$. Then, the posterior distribution is obtained with Bayes' theorem. The prediction is performed by integrating out θ in the likelihood function with respect to θ by using the posterior distribution.

In frequentist theory, as mentioned above, parameters are treated as unknown constants. There are various methods to estimate the parameter from observed data and the most widely used method would be to maximize the likelihood, which is called maximum likelihood estimation (see Section 2.1.1 for the detail). To evaluate the uncertainty in the frequentist theory, for example, a method called bootstrap is widely used for details (Bishop, 2006). In this method, first, we create new datasets from the original dataset by the resampling. Then, we estimate the parameter for each dataset. Finally, we obtain the uncertainty of the parameter as the variation of the estimated parameters across datasets. Since the parameter is a constant in the frequentist theory, a prediction is obtained by directly substituting the estimated parameter into the model. A schematic illustration of maximum likelihood estimation is shown in Figure. 1.3. In Figure. 1.3, θ is inferred by maximum likelihood estimation. The prediction is obtained by just substituting the estimated θ into the likelihood function.

Next, we describe the advantages of Bayesian inference. Bayesian inference has been used in real applications including social science and medical science recently increasingly (Bishop, 2006; Murphy, 2012). The reason is that Bayesian inference is useful for solving inverse problems. In inverse problems, when given the effects, our goal is to find a probability of the cause of them.

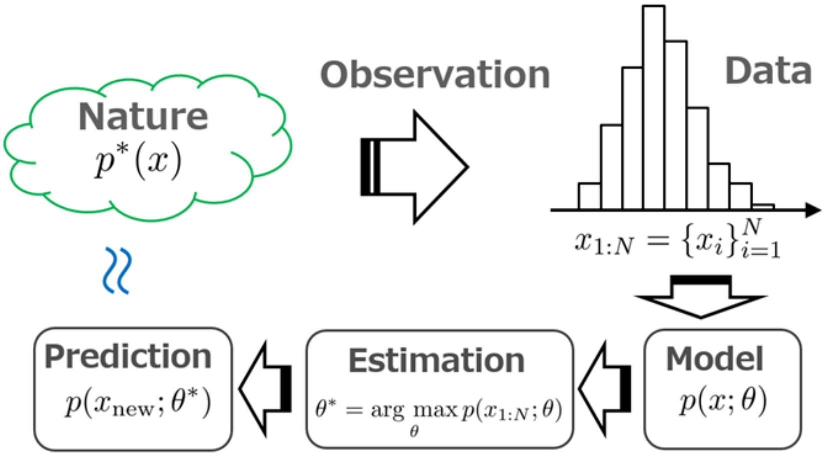

Nature
$p^*(x)$
Observation
Data
$x_{1:N} = \{x_i\}_{i=1}^N$
$\wr\wr$
Prediction
$p(x_{\text{new}};\theta^*)$
Estimation
$\theta^* = \arg\max_\theta p(x_{1:N};\theta)$
Model
$p(x;\theta)$

FIGURE 1.3: Schematic procedure of Maximum likelihood estimation.

We can estimate it by the posterior distribution in Bayesian inference. The other advantage is the evaluation of uncertainty about the model and the prediction. Uncertainty can be used as a criterion for model selection and the reliability criterion of the prediction (Bishop, 2006; Murphy, 2012). In particular about the prediction, in statistical inference, we usually assume that the past observed data and unknown future data are generated from the same data generating mechanism. However, it is difficult to verify whether this assumption holds in practice. For example, adversarial samples (Elsayed et al., 2018) seem almost the same as the past observed data for human eyes, but it is completely different from the past observed data for the model. So the prediction suffers from the severe breakdown. One approach tackling this problem is to clarify what types of observed data the model has already learned and for what kind of future data the model can make a reliable prediction. For this purpose, we can use uncertainty as the criterion of how reliable the prediction is and this approach showed promising results even for adversarial data (Li and Gal, 2017; Wang et al., 2019). Other practical advantages of Bayesian inference are that the use of a prior distribution prevents the overfitting and the probabilistic model is expressed as a graphical model, and thus it is intuitive and easy to interpret (Bishop, 2006; Wang and Yeung, 2016). Note that, when performing Bayesian inference, we need to carefully select the likelihood and a prior distribution based on the observed data and what kind of information we want to extract from the data.

Finally, we remark the difference of predictions between frequentist and Bayesian inference. Many studies have been conducted about the generalization ability of the frequentist theory and Bayesian inference (Konishi and Kitagawa, 1996; Fushiki et al., 2005; Shimodaira, 2000; Efron et al., 1998). However, it is difficult to say which prediction is superior in practical applications, because, in these theoretical studies, impractical assumptions are used, for example, the parameter in the frequentist model can be estimated accurately, the posterior distribution can be accurately evaluated, and the space of feasible models contain the true data generating mechanism. These assumptions are too strong and also too hard to confirm in practical settings. Therefore, it is important to select an appropriate inference method based on the problem one wants to handle. For example, when one wants to evaluate uncertainty or wants to solve inverse problems, Bayesian

Table 1.1: History of approximations in Bayesian inference.

Years	Methods
1763,1774	Birth of Bayes' theorem by Bayes and Laplace (Fienberg et al., 2006)
1930s to 40s	Formal description of Bayesian probability by Kolmogorov (Rukhin, 1990)
1930s	Development of basic ideas of Monte Carlo sampling in nuclear physics (Robert and Casella, 2011)
1939	Summary of the modern concepts of Bayesian inference (Jeffreys, 1939)
1949	Birth of Monte Carlo sampling (Metropolis and Ulam, 1949)
1953	Metropolis sampling (Metropolis et al., 1953)
1970	(Formal formulation) Metropolis-Hasting sampling (Hastings, 1970)
1972	The first work that proposed hierarchical Bayesian models (Lindley and Smith, 1972)
1977	Expectation-Maximization algorithm (Dempster et al., 1977)
1982	Belief propagation (Pearl, 1982)
1984	Gibbs sampling (Geman and Geman, 1984)
1987	Hybrid Monte Carlo (Duane et al., 1987)
1990	The first work using Markov chain Monte Carlo for the evaluation of the marginal probability (Gelfand and Smith, 1990)
1993	Sequential Monte Carlo (Gordon et al., 1993)
1994	Metropolis adjusted Langevin algorithm (Grenander and Miller, 1994)
1996(1993)	Variational inference (Saul et al., 1996; Jaakkola and Jordan, 2013; Hinton and Van Camp, 1993)
2001	Expectation propagation (Minka, 2001)
2011	Stochastic gradient Langevin dynamics (Welling and Teh, 2011)
2013	Stochastic variational inference (Hoffman et al., 2013)
2014	Black-box variational inference (Ranganath et al., 2014)
2019	Zig-zag subsampling (Bierkens et al., 2019)

inference is a promising approach. For these reasons, this dissertation focuses on Bayesian inference.

1.3 Approximate Bayesian inference

Here, we describe the drawback of Bayesian inference and its solution. The biggest problem of Bayesian inference is that the posterior and predictive distributions cannot often be obtained analytically and we need to evaluate them numerically. Moreover, since practical models are usually high-dimensional, it is difficult to obtain the exact numerical evaluation due to high computation costs. Therefore, we need approximation methods for them. Bayesian inference has a long history, but its practical deployment began only a few decades ago because of the necessity of computation power and approximation methods. We summarized the history of approximations in Bayesian inference in Table 1.1 (Fienberg et al., 2006; Robert and Casella, 2011). Originally, Bayes' theorem was derived in the 18th century. The modern concepts of Bayesian inference were established until 1939. However, the first practical hierarchical models were proposed in 1972 and also the first numerical approximation was performed in 1990. More than a half century passed from the establishment of the concepts of Bayesian inference when Bayesian inference became practical. The recent success of Bayesian inference is mainly due to the advances of numerical calculators and the development of approximation methods such as sampling methods or parametric approximation techniques. Therefore, developing better approximation methods is essential for Bayesian inference.

Currently, various approximation methods have been developed (see Chapter 2 for details). We need to select an appropriate approximation method in consideration of the characteristics of the model and the data we use. Thus, we need to consider an appropriate combination of a model, a prior distribution and an approximation method in practice.

Approximation methods are categorized into two types in general (Bishop, 2006; Murphy, 2012). One is the parametric approximation and the other is the sampling-based approximation. The advantage of the parametric approximation is that the parametric assumptions make the algorithm computationally tractable and can be applied to high-dimensional problems. The disadvantage is that since we express the complex posterior distribution by a simpler parametric distribution, it is usually too restrictive to express the posterior distribution exactly, and thus, the obtained approximate distribution could be far from the true posterior distribution in general. Moreover, it is difficult to bound the magnitude of this difference theoretically in many situations. On the other hand, the approximation by sampling can approximate any posterior distribution precisely if a sufficient number of samples is used. However, especially in the case of a high-dimensional or multimodal distribution, the computation cost to draw samples becomes very large. Depending on these advantages, disadvantages, the complexity of the model, and the accuracy we want to achieve, we need to design a suitable approximation method.

Finally, we describe the meaning of Bayesian inference in the age of deep learning. One of the advantages of traditional Bayesian inference is that we can incorporate our prior knowledge or assumptions into a model as a prior distribution. It is known empirically that Bayesian inference shows more trustworthy results than the traditional maximum likelihood estimator for the small sample size problems by choosing the appropriate prior distribution (McNeish, 2016). By using nonparametric methods, we can construct flexible models even if solid assumptions for the prior distribution are not available. Another interesting relation is a combination of deep learning and Bayesian inference. Such combinations are known as deep Bayesian inference or deep graphical models (Johnson et al., 2016; Wang and Yeung, 2016). Those methods can enjoy strong feature extraction properties of deep learning and the structured representation power of Bayesian inference. Optimization algorithms for those methods have been developed based on the traditional approximation methods in Bayesian inference.

In this dissertation, we discuss the approximation techniques for Bayesian inference focusing on outliers in the observed data. We first present two approaches about outlier robust inference and corresponding approximation methods and then we finally present a new approximation approach that has the hybrid nature of the parametric and sampling approach.

1.4 Robust inference

Here, we discuss robustness to outliers. By robustness, we mean that "*an insensitivity to small deviations from the assumptions*", following the seminal book (Huber and Ronchetti, 2011). When we perform statistical inference, there are various assumptions, such as the i.i.d. (independent identically distributed) assumption, distributional assumptions, and assumptions about the prior knowledge. We say that an estimation method is robust if it is not sensitive to a slight deviation from these assumptions.

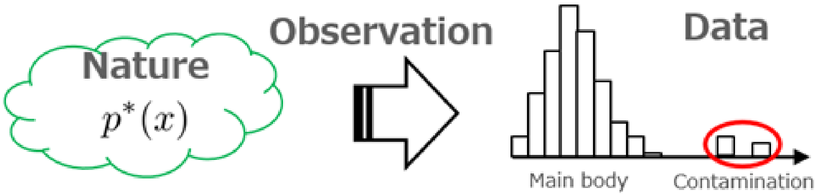


FIGURE 1.4: Schematic procedure of outliers.

Robustness we consider in this dissertation is related to the assumption of distributions. This is the case when there is an abnormality in the data generating mechanism of the observed data. In addition to the data that we want to analyze, unrelated contamination is added to the observed data somehow. Usually, the data we want to analyze is the main body and the proportion of contamination is small. Such contamination does not necessarily reflect the property we want to analyze and behaves very differently from the other main body of the data. Such data are called outliers. Although the proportion of outliers is small, outliers are usually located to a tail in the empirical distribution of the observed data, they have a significant effect on the estimation results. Robustness to outliers is getting more important these days since recent advances in sensor technology give a vast amount of data with spiky noise and crowd-annotated data is full of human errors (Raykar et al., 2010; Zhang et al., 2016; Liu et al., 2012; Bonald and Combes, 2017). A schematic description of outliers is shown in Figure. 1.4. Compared to Figure. 1.3, there is contamination which are marked by a red circle in the tail of the empirical distribution. We are not interested in such contamination is statistical inference.

If an estimation method is not robust against outliers, the prediction will be strongly influenced by outliers. In particular, when there is even one outlier in the observed data and it is located infinitely far away from the main body of the data, it can have an infinite effect on the estimation results. Thus, the development of an algorithm that is robust against outliers is a very important problem in actual application.

For such outliers, we can apply a two-step method, that is, first we remove outliers from the observed data, and then perform an ordinary estimation method assuming that the remaining data do not have outliers anymore. However, this approach has several drawbacks (Huber and Ronchetti, 2011). First, there are two sources of bias and variance: removing outliers in the first stage and the estimation of the parameter in the second stage. Second, removing outliers under multi-variable models is a difficult problem itself. Even if we successfully eliminate outliers from the observed data, the remaining data no longer satisfies the assumption of i.i.d., and it will be difficult to perform theoretical analysis for such data (Huber and Ronchetti, 2011). Therefore, instead of this two-step approach, developing robust inference which automatically eliminates the effect of outliers is more important. A schematic description of robust inference is shown in Figure. 1.5. We want to eliminate the effect of contamination which is marked by a red circle. We want to make a model which captures the information of the main body of the observed data marked by a dotted line.

Important concepts in robust inference are efficiency, stability, and breakdown (Huber and

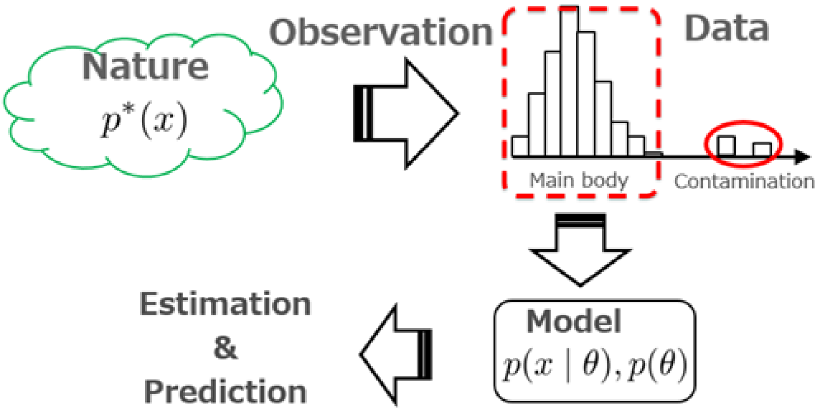


FIGURE 1.5: Schematic procedure of Robust inference.

Ronchetti, 2011). Among them, the concept which is unique to robust inference is stability and breakdown. Stability means that the estimation method is not sensitive to small deviations from assumptions, which we have already described above as the definition of robustness. Breakdown means that even if a small number of outliers have an infinitely large deviation from the majority of the data, the estimation will not fail. About the efficiency, various theoretical results have been obtained in existing researches (Van der Vaart, 1998; Huber and Ronchetti, 2011). In this dissertation, we focus on the concept of breakdown.

So far, many methods to enhance robustness against outliers have been proposed (Huber and Ronchetti, 2011). One of the most widely used approaches is a model-based approach. For example, assume that the model uses the Gaussian distribution. Since outliers often appear in the tails of the empirical distribution of the observed data and the Gaussian distribution has a short tail, the result of estimation is strongly affected by outliers. Thus, the Gaussian distribution is not favorable in terms of robustness. In the model-based approach, we replace the Gaussian distribution with a distribution which has a long tail so that the model is less affected by outliers. One example is the Student-t distribution, which has a very similar shape to the Gaussian distribution but has a longer tail.

1.5 Contributions

Here, we describe the contributions of this dissertation.

1.5.1 Expectation propagation for t-exponential family using q-algebra

Although the Student-t distribution is favorable for robustness, it is difficult to handle compared to the Gaussian distribution as a component of probabilistic models. The advantage of using the Gaussian distribution is that their moments, conditional distribution, and joint distribution can be computed analytically and it is a member of the exponential family. Thus, the calculation can be performed efficiently through natural parameters. On the other hand, the Student-t distribution is not a member of this family, thus we cannot utilize the useful properties to develop computationally efficient approximation methods. To address this problem, we borrow the mathematical tools of *q-algebra* (Nivanen et al., 2003; Suyari and Tsukada, 2005). from statistical physics and show that the *pseudo additivity* of distributions under the q-algebra allows us to perform the calculation of the Student-t distributions through natural parameters. We then develop an *expectation propagation* (EP) algorithm for the Student-t distributions, which provides a deterministic approximation to the posterior or predictive distribution with simple moment matching. We finally apply the proposed EP algorithm to the *Bayes point machine* (Minka, 2001) and *Student-t process classification*, and demonstrate their performance numerically.

1.5.2 Variational inference based on robust divergences

While replacing a model to a heavy-tailed one (e.g., from the Gaussian distribution to the Student-t distribution) is a standard approach to enhance robustness, it can only be applied to simple models. Exploring the choice of the replacement to find the robust model is not a promising approach since we need a large computation cost for the estimation of complex models. Thus a systematic approach to enhance robustness is required.

To address this problem, based on Zellner's optimization and variational formulation of Bayesian inference (Zellner, 1988), we propose an outlier-robust pseudo-Bayesian variational method by replacing the Kullback-Leibler divergence used for data fitting in the reformulated Bayes' theorem to a robust divergence such as the β- and γ-divergences (Basu et al., 1998; Fujisawa and Eguchi, 2008). With these divergences, we can automatically ignore outliers since the weights of outliers become small compared to ordinary data points. An advantage of our approach is that superior but complex models such as deep networks can also be handled. We theoretically prove that, for deep networks with ReLU activation functions, the influence function in our proposed method is bounded, while it is unbounded in the ordinary variational inference. This implies that our proposed method is robust to both input and output outliers, while the ordinary variational method is not. We experimentally demonstrate that our robust variational method outperforms ordinary variational inference in regression and classification with deep networks.

1.5.3 Bayesian posterior approximation via greedy particle optimization

The above proposed robust inference methods are parametric approximation methods, and therefore, bias from the true distribution can occur and it is hard to evaluate this bias theoretically in general. As we described above, there are advantages and disadvantages between the parametric approximation and sampling-based approximation. Based on this, we aimed to develop a method that combines the advantages of each approximation, that is, flexibility and the theoretical guarantee of the sampling-based approach and computational efficiency of the parametric approach. We developed an algorithm that the posterior distribution is approximated by an empirical distribution as the sampling-based approach and the points of the empirical distribution are estimated by solving an optimization problem. Specifically, we proposed minimizing a distance measure called Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) by using the Frank-Wolfe algorithm (Jaggi, 2013). The distance between the approximate posterior distribution and the true posterior distribution is minimized as a constrained convex optimization problem on the reproducing kernel Hilbert space (RKHS). Based on this approach, the obtained algorithm is computationally efficient and can be applied to high-dimensional problems, and the approximation quality is theoretically guaranteed.

1.6 Organization

Finally, this dissertation is organized into 6 chapters. In Chapter 2, we review the basics of Bayesian inference and some widely used approximation methods. In addition to that, we introduce the concept of robust inference. In Chapter 3, we present the computationally efficient algorithm for the long-tailed distribution by using the q-algebra. In Chapter 4, we discuss the systematic robust inference by changing the distance measure rather than changing the model. In Chapter 5, we introduce a novel approximation strategy which combines the parametric and sampling approximation method. Finally, in Chapter 6, we discuss the summary and future developments.

Chapter 2

Preliminaries

In this chapter, we introduce the basics of approximate Bayesian inference. Then we introduce the basics of outlier robust inference.

2.1 Basics of Bayesian inference

Here, we first describe a general formulation of Bayesian inference. Then we provide a simplified formulation for the exponential family.

2.1.1 Formulation

Let us consider the problem of estimating an unknown probability distribution $p^*(x)$ from its independent samples $x_{1:N} = \{x_i\}_{i=1}^N$. We also use D to express the observed data for simplicity.

To this end, we consider a parametric model $p(x|\theta)$ with the parameter $\theta \in \mathbb{R}^d$. This $p(x|\theta)$ is called the likelihood function and expresses the plausibility of the observed data depending on the choice of the parameter θ. If samples $x_{1:N} = \{x_i\}_{i=1}^N$ are independent of each other, we can express the likelihood as $\prod_{i=1}^N p(x_i|\theta) = p(D|\theta)$. In Bayesian inference, the parameter θ is regarded as a random variable, having the prior distribution $p(\theta)$ which expresses our belief or assumption about θ before observing the data. Then, we incorporate the information of the observed data into the parameter by using Bayes' theorem. With Bayes' theorem, we obtain the posterior distribution $p(\theta|D)$ which is defined as

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}, \tag{2.1}$$

where $p(D) = \int p(D|\theta)p(\theta)d\theta$ is called the marginal likelihood. Thus, given data $\{x_i\}_{i=1}^N$, we express all the uncertainty thorough the probability distribution over the parameter. When new data is given, we evaluate the uncertainty of the data by using the predictive distribution,

$$p(x_{\text{new}}|D) = \int p(\theta|D)p(x_{\text{new}}|\theta)d\theta, \tag{2.2}$$

where the parameter is integrated. Thus, calculating the posterior distribution is the central task in Bayesian inference.

Let us compare Bayesian inference with maximum likelihood (ML) estimation. In ML estimation, we minimize the error measured by the Kullback-Leibler(KL) divergence D_{KL} from $p^*(x)$ to $p(x;\theta)$:

$$D_{\mathrm{KL}}\left(p^*(x) \| p(x;\theta)\right) = \int p^*(x) \log \left(\frac{p^*(x)}{p(x;\theta)} \right) dx. \tag{2.3}$$

Note that in ML estimation, θ is not a random variable. Compared to Bayesian inference, ML estimation provides a point estimate of the parameter. Since $p^*(x)$ is unknown in practice, it is replaced with

$$D_{\mathrm{KL}}\left(\hat{p}(x) \| p(x;\theta)\right) = \mathrm{Const.} - \frac{1}{N} \sum_{i=1}^{N} \ln p(x_i;\theta), \tag{2.4}$$

where $\hat{p}(x) = \frac{1}{N} \sum_{i=1}^{N} \delta(x, x_i)$ is the empirical distribution and δ is the Dirac delta function. Thus, we minimize this empirical KL divergence to estimate the parameter. Equating the partial derivative of Eq.(2.4) with respect to θ to zero, we obtain the following estimating equation:

$$0 = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial}{\partial \theta} \ln p(x_i;\theta). \tag{2.5}$$

By solving this equation, we get the ML estimate of θ.

Let us go back to Bayesian inference and re-interpret the posterior distribution as an optimization problem. Zellner (1988) showed that the posterior distribution $p(\theta|D)$ can also be obtained by solving the following optimization problem: [1]

$$\operatorname*{arg\,min}_{q(\theta) \in \mathcal{P}} L(q(\theta)), \tag{2.6}$$

where $\mathcal{P}$ is the set of all probability distributions, $-L(q(\theta))$ is the *evidence lower-bound* (ELBO),

$$L(q(\theta)) = D_{\mathrm{KL}}(q(\theta) \| p(\theta)) - \int q(\theta) \left(-N d_{\mathrm{KL}}\left(\hat{p}(x) \| p(x|\theta)\right)\right) d\theta, \tag{2.7}$$

and $d_{\mathrm{KL}}\left(\hat{p}(x) \| p(x|\theta)\right)$ denotes the *cross-entropy*:

$$d_{\mathrm{KL}}\left(\hat{p}(x) \| p(x|\theta)\right) = -\frac{1}{N} \sum_{i=1}^{N} \ln p(x_i|\theta). \tag{2.8}$$

Note that the posterior distribution Eq.(2.1) can be expressed as

$$p(\theta|D) = \frac{e^{-N d_{\mathrm{KL}}(\hat{p}(x) \| p(x|\theta))} p(\theta)}{p(D)}. \tag{2.9}$$

This reformulation allows us to understand Bayesian inference as the minimization problem of the cross entropy between a parametric model and the true data generating distribution with a

[1] Zellner's formulation of Bayesian inference was also used for extending variational inference to constrained methods (Zhu et al., 2014; Koyejo and Ghosh, 2013).

regularization term. Compared to ML estimation, the dependency of θ in the objective function is integrated out. This reformulation plays an important role in developing an approximation method for the posterior distribution.

2.1.2 Exponential family

In statistical inference, an exponential family distribution is widely used as the likelihood function due to the fact that their calculation can be performed efficiently and analytically through natural parameters or sufficient statistics (Bishop, 2006). This property is particularly useful in Bayesian inference since we can obtain an analytical expression of the posterior distribution if the corresponding conjugate prior is used as the exponential family likelihood.

An exponential family is defined as

$$p(x;\theta) = \exp(\langle \Phi(x), \theta \rangle - g(\theta)), \tag{2.10}$$

where $\langle \cdot, \cdot \rangle$ means the inner product on $\mathbb{R}^d$, $\Phi(x)$ is some function of x, and $g(\theta)$ is the normalizing constant. The parameter θ is called the natural parameter. Many distributions which are used in Bayesian inference are categorized into this family. The most famous example is the Gaussian distribution which is defined as

$$N(x;\mu,\Sigma) = \frac{1}{\sqrt{\det 2\pi\Sigma}} \exp\Big\{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\Big\}, \tag{2.11}$$

where μ, Σ are its mean vector and covariance matrix and $\det$ means the determinant of a matrix. The Gaussian distribution is the most widely used because its moments, conditional distribution, and joint distribution can be computed analytically. As a member of the exponential family, its natural parameter and $\Phi(x)$ are expressed as

$$\theta = \begin{pmatrix} \Sigma^{-1}\mu \\ -\frac{1}{2}\Sigma^{-1} \end{pmatrix}, \quad \Phi(x) = \begin{pmatrix} x \\ xx^\top \end{pmatrix}. \tag{2.12}$$

Other examples can be found in Bishop (2006). A useful property of the exponential family is that the expectation of the sufficient static is equal to the gradient of the normalizing constant (Bishop, 2006):

$$\mathbb{E}_p[\Phi(x)] = \nabla_\theta g(\theta), \tag{2.13}$$

where $\mathbb{E}_p$ denotes the expectation with respect to $p(x;\theta)$. Another important property is that the product of densities from the same exponential family results in an unnormalized density which is a member of the same exponential family:

$$\exp(\langle \Phi(x), \theta_1 \rangle - g(\theta_1)) \exp(\langle \Phi(x), \theta_2 \rangle - g(\theta_2)) = \exp(\langle \Phi(x), (\theta_1 + \theta_2) \rangle - \tilde{g}(\theta_1, \theta_2)), \tag{2.14}$$

where $\tilde{g}$ is the normalizing constant. These properties are important in developing an estimation method or an approximation method.

Next, we discuss how an exponential family is used in statistical inference. When we use the exponential family as the likelihood function, it is written as

$$p(D;\theta) = \exp(\langle \sum_{i=1}^{N} \Phi(x_i), \theta \rangle - Ng(\theta)). \tag{2.15}$$

We consider ML estimation for θ and we take the logarithm and derivative with respect to θ,

$$\nabla_\theta \log p(D;\theta) = \sum_{i=1}^{N} \Phi(x_i) - N\nabla_\theta g(\theta)), \tag{2.16}$$

Then applying Eq.(2.13) and setting the left-hand side equal to 0, we obtain

$$0 = \frac{1}{N} \sum_{i=1}^{N} \Phi(x_i) - \mathbb{E}_p[\Phi(x)]. \tag{2.17}$$

This is called the moment matching property of the exponential family (Bishop, 2006). As we can see, the solution of ML estimation for the exponential family only depends on $\Phi(x)$. Thus, $\sum_{i=1}^{N} \Phi(x_i)$ is often called the sufficient static.

To conclude this section, we describe the relation of the exponential family to Bayesian inference. When the likelihood function is given by

$$p(D|\theta) \propto \exp(\langle \sum_{i=1}^{N} \Phi(x_i), \theta \rangle) = \exp(\langle N\bar{\Phi}, \theta \rangle), \tag{2.18}$$

as a prior distribution, we use the conjugate prior distribution which is defined as

$$p(\theta; \tau_0, \Phi_0) \propto \exp(\langle \tau_0 \Phi_0, \theta \rangle). \tag{2.19}$$

Then, we get the posterior distribution as

$$\begin{aligned} p(\theta|D) &\propto p(D|\theta)p(\theta) \\ &\propto \exp(\langle (\tau_0\Phi_0 + N\bar{\Phi}), \theta \rangle) \\ &= \exp(\langle \frac{\tau_0\Phi_0 + N\bar{\Phi}}{\tau_0 + N}(\tau_0 + N), \theta \rangle) \\ &= p(\theta; \tau_0 + N, \frac{\tau_0\Phi_0 + N\bar{\Phi}}{\tau_0 + N}). \end{aligned} \tag{2.20}$$

This means that the posterior distribution is expressed by the same function form as the conjugate prior distribution and the parameter of the conjugate prior distribution is updated by the likelihood function. Thus, by using the conjugate prior distribution, we obtain an analytical expression for the corresponding posterior distribution. When the Gaussian distribution is used for the likelihood function, the conjugate prior distribution of the mean variable is the Gaussian and that of the variance variable is the inverse gamma distribution (Bishop, 2006).

2.2 Approximate Bayesian inference

In the previous section, we reviewed the basics of Bayesian inference. When we apply Bayesian inference in real-world problems, the difficult point is that we cannot evaluate the posterior distribution exactly in many cases since the likelihood and the prior distribution do not satisfy the conjugate relation as we reviewed in the previous section. Thus, we need an approximation method for the posterior distribution. Developing an appropriate approximation algorithm for the posterior distribution given the likelihood and the prior distribution is the central task in Bayesian inference in practice. Here, we introduce several approximation methods which are used widely in practice.

2.2.1 Variational inference (VI)

Variational inference (VI) is one of the most widely used approximation methods (Blei et al., 2017). VI approximates the posterior distribution with the parametric distribution from which we can easily draw samples. An exponential family is often used as the approximate distribution.

The procedure of VI is as follows: first, we prepare a parametric approximate distribution such as an exponential family and then, we estimate the parameter of the approximate distribution by minimizing a "distance" between the posterior distribution and the approximate distribution. The most widely used distance is the KL divergence. Thus, VI can be interpreted as the minimization problem of the KL divergence. Let us express the parametric approximate distributions as $q(\theta;\lambda) \in \mathcal{Q}$, where λ is the parameter to be optimized. Then, the optimization problem is written as

$$\mathop{\arg\min}_{q(\theta;\lambda)\in\mathcal{Q}} D_{\mathrm{KL}}(q(\theta;\lambda)\|p(\theta|D)), \tag{2.21}$$

where $p(x|D)$ is the true posterior distribution.

We can reformulate the above minimization problem by using the marginal log-likelihood $\ln p(D)$ as

$$\begin{aligned} D_{\mathrm{KL}}\left(q(\theta;\lambda)\|p(x|D)\right) &= \int q(\theta;\lambda)\log\left(\frac{q(\theta;\lambda)}{p(\theta|D)}\right)d\theta \\ &= \int q(\theta;\lambda)\log\left(\frac{q(\theta;\lambda)}{p(D|\theta)p(\theta)/p(D)}\right)d\theta \\ &= \log p(D) - \int q(\theta;\lambda)\log\left(\frac{p(D|\theta)p(\theta)}{q(\theta;\lambda)}\right)d\theta. \end{aligned} \tag{2.22}$$

Thus, minimizing the KL divergence is equivalent to maximizing the second term of Eq.(2.22). The second term is called the evidence lower bound (ELBO) which is equivalent to Eq.(2.7). Since the

KL divergence is always non-negative, we get

$$\begin{aligned}\log p(D) &= \int q(\theta;\lambda) \log\left(\frac{p(D|\theta)p(\theta)}{q(\theta;\lambda)}\right) d\theta + D_{\mathrm{KL}}\left(q(\theta;\lambda) \| p(\theta|D)\right) \\ &\geq \int q(\theta;\lambda) \log\left(\frac{p(D|\theta)p(\theta)}{q(\theta;\lambda)}\right) d\theta := L(\lambda). \end{aligned} \tag{2.23}$$

We can confirm that the objective function of VI is $L(\lambda)$ which is upper bounded by the marginal log-likelihood and the bound is tight when the approximate distribution is equivalent to the true posterior distribution. In conclusion, the optimization problem we solve becomes

$$\underset{q(\theta;\lambda)\in\mathcal{Q}}{\arg\min}\, L(q(\theta;\lambda)). \tag{2.24}$$

In comparison with Eq.(2.6), which is the optimization formulation of the posterior distribution with its domain being all densities, the domain of Eq.(2.24) is restricted to the prepared parametric distributions.

2.2.2 Assumed density filtering (ADF) and expectation propagation (EP)

In Section 2.1.1, we introduced VI which minimizes the KL divergence. In VI, we approximate the posterior distribution with a single parametric distribution $q(\theta;\lambda)$. Here, we consider a different parametric approximate distribution which captures each factor of $p(D|\theta) = \prod_i p(x_i|\theta)$ in the following way. For simplicity, we express the likelihood function for the i-th data as $l_i(\theta)$. The total likelihood is given as $\prod_{i=1}^N l_i(\theta)$ and the posterior distribution is expressed as $p(\theta|D) \propto p(\theta)\prod_i l_i(\theta)$. Then, we prepare the parametric approximate posterior distribution as the product of data-corresponding terms as

$$\widetilde{p}(\theta) = \frac{1}{Z}\prod_i \widetilde{l}_i(\theta), \tag{2.25}$$

where Z is the normalizing constant. In the above expression, the factors $\widetilde{l}_i(\theta)$, which are often called the *site approximations* (Seeger, 2005), correspond to the local likelihood $l_i(\theta)$. An exponential family distribution is often used as the site approximations. Then we minimize the reverse KL divergence $D_{\mathrm{KL}}\left(p(\theta|D) \| \widetilde{p}(\theta)\right)$. This reverse KL divergence and the product form of the approximate posterior distribution can capture the different properties of the true posterior distribution than VI (Bishop, 2006). We review two major algorithms to optimize $D_{\mathrm{KL}}\left(p(\theta|D) \| \widetilde{p}(\theta)\right)$. They are known as assumed density filtering (ADF) and expectation propagation (EP) (Minka, 2001).

ADF is online approximation method for the posterior distribution. Suppose that $i-1$ data $x_1 \ldots, x_{i-1}$ have already been processed and an approximation to the posterior distribution, $\widetilde{p}_{i-1}(\theta)$, has already been obtained. Given i-th data x_i, the posterior distribution $p_i(\theta)$ can be obtained as

$$p_i(\theta) \propto \widetilde{p}_{i-1}(\theta) l_i(\theta). \tag{2.26}$$

Since the posterior distribution $p_i(\theta)$ cannot be obtained analytically, it is approximated by minimizing the reverse KL divergence from $p_i(\theta)$ to its approximation:

$$\widetilde{p}_i(\theta) = \arg\min_{\widetilde{p}} D_{\mathrm{KL}}(p_i(\theta) \| \widetilde{p}(\theta)). \tag{2.27}$$

Note that if p_i and $\widetilde{p}$ are both exponential family members, the optimization problem Eq.(2.27) is reduced to moment matching (Bishop, 2006).

Although ADF is an effective method for online learning, it is not favorable for batch (i.e., not online) learning because the approximation quality depends heavily on the permutation of data (Minka, 2001). To overcome this problem, EP was proposed (Minka, 2001). Contrary to ADF, the EP algorithm is an effective method when the whole data is given in advance. EP updates the site approximations iteratively with the following four steps.

1. First, when we update site $\widetilde{l}_j(\theta)$, we eliminate the effect of site j from the total approximation as

$$\widetilde{p}^{\backslash j}(\theta) = \frac{\widetilde{p}(\theta)}{\widetilde{l}_j(\theta)}, \tag{2.28}$$

   where $\widetilde{p}^{\backslash j}(\theta)$ is often called a *cavity distribution* (Seeger, 2005). If an exponential family distribution is used, the above calculation is reduced to subtraction of natural parameters.

2. Second, we incorporate likelihood $l_j(\theta)$ by minimizing the divergence $D_{\mathrm{KL}}(\widetilde{p}^{\backslash j}(\theta) l_j(\theta)/Z^{\backslash j} \| \widetilde{p}(\theta))$, where $Z^{\backslash j}$ is the normalizing constant. Note that this minimization is reduced to moment matching for an exponential family distribution. After this step, we obtain $\widetilde{p}(\theta)$.

3. Third, we exclude the effect of terms other than j, which is equivalent to calculating a cavity distribution as $\widetilde{l}_j(\theta)^{\mathrm{new}} \propto \frac{\widetilde{p}(\theta)}{\widetilde{p}^{\backslash j}(\theta)}$.

4. Finally, we update the site approximation by replacing $\widetilde{l}_j(\theta)$ by $\widetilde{l}_j(\theta)^{\mathrm{new}}$.

It should be noted that calculations of EP are reduced to addition or subtraction of natural parameters for an exponential family distribution and it is computationally efficient.

2.2.3 Markov chain Monte Carlo

The drawback of the parametric approximation method is that the parametric assumption is often too restrictive to approximate the true posterior distribution, and therefore the approximate distribution never converges to the posterior distribution and no theoretical guarantee is assured in general.

Here, we review the approximation of the posterior by a set of points $\{\theta_n\}_{n=1}^N$, $\hat{p}(\theta) = \sum_{n=1}^N \delta(\theta, \theta_n)/N$, where N is the number of points. This approximation is more expressive than the parametric approximation since no parametric assumptions are required. The Monte Carlo (MC) method is typically used to obtain the points $\{\theta_n\}_{n=1}^N$, that is, we draw $\{\theta_n\}_{n=1}^N$ from the posterior distribution randomly and independently (Bishop, 2006).

There are various types of MC methods, and here, we only review the stochastic gradient Langevin dynamics (SGLD) algorithm (Welling and Teh, 2011) since it is one of the most widely used MC methods in Bayesian inference. This method is an extension of the Langevin dynamics to the stochastic gradient, which enjoys the scalability with respect to the number of data. First, we introduce some notations. D represents full data, which is independent and identically distributed (i.i.d.), and it can be decomposed to subsets of data as $D = \cup_{q=1}^{|D|}\mathcal{D}_q$. Thus, we can write the likelihood function as $p(D|\theta) = \prod_q p(\mathcal{D}_q|\theta)$. We define the potential of the posterior distribution as

$$\tilde{U}(\theta) = -\log p(\theta|D), \tag{2.29}$$

and

$$U(\theta|\mathcal{D}_q) := -\log p(\theta|\mathcal{D}_q). \tag{2.30}$$

Then, the full potential can be expressed as the summation, $\tilde{U}(\theta) = \sum_q U(\theta|\mathcal{D}_q)$. In SGLD, instead of the full gradient, the stochastic gradient which uses a randomly chosen subset of data at each iteration is used. We express the stochastic potential at time t as

$$U_t(\theta) = \frac{1}{B_t}\sum_{q\in\mathcal{I}_t} U(\theta|\mathcal{D}_q) \tag{2.31}$$

, where $\mathcal{I}_t$ is a random subset of $[1, 2, \ldots, |D|]$ with size B_t. Based on these notations, the SGLD algorithm works as

$$d\theta_t = -\beta^{-1}\nabla U_t(\theta_t)dt + \sqrt{2\beta^{-1}}dw_t, \tag{2.32}$$

where $(w_t)_{t\geq 0}$ is a $\mathbb{R}^d$-valued Wiener process (Bakry et al., 2013). It is known that Eq.(2.32) has the stationary distribution $p(\theta|D)$ if the dynamics of Eq.(2.32) is ergodic. Thus, we can get the samples from the true posterior by using this dynamics.

To implement the SGLD algorithm, we need to discretize the above continuum stochastic differential equation (SDE). When we use the Euler-Maruyama scheme (Bakry et al., 2013) with a step size $h > 0$, we can implement the SGLD algorithm at the l-th iteration as

$$\theta_{(l+1)h} = \theta_{lh} - h\beta^{-1}\nabla U_l(\theta_{lh}) + \sqrt{2h\beta^{-1}}\epsilon_l, \quad \epsilon_l \sim N(0, I). \tag{2.33}$$

The purpose of the SGLD algorithm is to approximate the posterior average for a test function $f(\theta)$, $\bar{f} = \int f(\theta)p(\theta|D)d\theta$. Let us suppose that we get L samples $\{\theta_{lh}\}_{l=1}^{L}$ by running Eq.(2.33). Then, we approximate $\bar{f}$ with the ergodic average as $\hat{f} = \frac{1}{L}\sum_{l=1}^{L} f(\theta_{lh})$. It is known that the SGLD algorithm decreases the KL divergence between the true posterior and the distribution of the algorithm at each time step. Other types of sampling method such as Gibbs sampling also decrease the KL divergence between the current state and the posterior distribution.

2.2.4 Comparison of the approximation methods

In Section 2.2, we reviewed widely used approximation methods in Bayesian inference. In general, those approximation methods are categorized into two types. One is parametric approximations such as VI, ADF, and EP. In these methods, we approximate the posterior distribution with a parametric distribution and the parameter is estimated by solving the optimization problems as we reviewed. In many common Bayesian models, these parametric assumptions are often too restrictive to approximate the true posterior distribution exactly. Thus, the obtained approximation is biased from the true posterior distribution even if we solve the optimization problem exactly. The advantage of these parametric approximation methods is that they work well in practical Bayesian models which are usually high dimensional.

The other approximation approach is the sampling method such as MC, which approximates the posterior distribution by a finite set of points and these points are generated by random sampling. If we have a large number of samples, we can approximate the posterior distribution precisely. The drawbacks of this approach are that the vast computational cost is required to obtain samples from multi-modal and high-dimensional distributions and it is hard to decide when to stop the algorithm in practice.

Hence, there is a difference in terms of the approximation accuracy and computational cost between the parametric approximation and the sampling-based approximation. We need to choose an appropriate approximation method for a given Bayesian model based on these properties of the approximation methods. Let us consider a Bayesian neural network as an example. This model has a vast number of parameters, and therefore it is impossible to apply the sampling-based methods. Instead, we should use the parametric approximation since it can work well in high dimensional models. We also need to specify which parametric distribution we use and which objective function we minimize. We should select such combination based on what properties of the posterior distribution we want to capture.

In conclusion, Bayesian inference in practice is the combination of the choice of the likelihood function, the prior distribution, and the approximation method. These combinations should be determined based on what kind of data we treat and what kind of information we want to extract from data and how much we can tolerate as a computational burden.

2.3 Robust inference

Here, we briefly review the notion of robustness and its relation to Bayesian inference.

2.3.1 Robustness and outliers

Robustness is a fundamental topic in machine learning and statistics (Huber and Ronchetti, 2011; Murphy, 2012). Although a specific definition of robustness may be problem-dependent, a commonly shared notion is "*an insensitivity to small deviations from the assumptions*", according to the seminal book by Huber and Ronchetti (2011). Although robustness to outliers is a classic problem, it is getting more important these days since recent advances in sensor technology give a vast amount of data

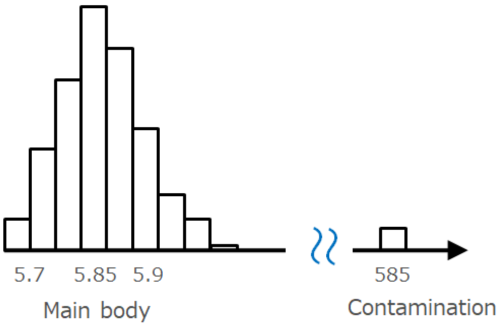


FIGURE 2.1: Schematic example of an outlier

with spiky noise and crowd-annotated data is full of human errors (Raykar et al., 2010; Zhang et al., 2016; Liu et al., 2012; Bonald and Combes, 2017).

In this thesis, we simply refer to outliers as the data which are not the main body of the data. Figure.2.1 shows a schematic view of outliers. This figure illustrates the empirical distribution and an outlier which locates far away from the main body of the data. In practice, we want to extract the information about the main body of the data and outliers are regarded as contamination to the main body of the data. Thus, we want to reduce the effect of contamination since it does not reflect the information that we are interested in. In conclusion, the objective of outlier robust inference is to eliminate the effect of outliers automatically and extract the information only from the main body of the data. Note that, in the field of anomaly detection, we actively try to discover outliers (Bishop, 2006).

Let us state the above intuition formally. We assume that the observed data are generated from $\{x_i\}_{i=1}^N \overset{\text{i.i.d.}}{\sim} p^*(x)$. Let P be an empirical measure of the observed data $\{x_i\}_{i=1}^N$:

$$P\left(x\right) = \frac{1}{N}\sum_{i=1}^{N}\Delta_{x_i}\left(x\right), \tag{2.34}$$

where $\Delta_{x_i}(x)$ stands for a point-mass 1 at x_i and also let $P_{\varepsilon,z}$ be a contaminated version of P at z:

$$P_{\varepsilon,z}(x) = (1-\varepsilon)P(x) + \varepsilon\Delta_z(x), \tag{2.35}$$

where ε is the contamination proportion. This means that there exists contamination at a point z. We also express the contaminated version of the corresponding density as

$$p^*(x) = (1-\varepsilon)p_0^*(x) + \varepsilon\delta(x,z),$$

where $p_0^*(x)$ expresses the density of the main body data. We aim at placing an estimated probability, e.g., $p(x;\theta)$ close to the main body of the unknown density $p_0^*(x)$. Figure.2.2 shows a schematic picture about this.

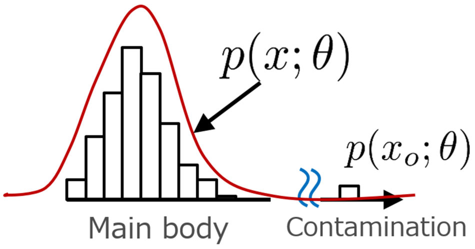


FIGURE 2.2: Schematic example of an outlier

2.3.2 Robustness and Bayesian inference

Here, we will explain a standard approach for outlier robust inference in Bayesian inference (Bishop, 2006; Murphy, 2012). Let us consider a regression problem as an example. Given pairs of inputs and outputs $D = \{(x_i, y_i)\}_{i=1}^{N}$, we assume a function $y_i = f(x_i)$. We infer a distribution over the function f, that is, $p(f|D)$. We use a Gaussian process (GP) regression model as a probabilistic model(Rasmussen and Williams, 2006). A GP is a typical Bayesian method based on the Gaussian distribution, which is used for various purposes such as regression, classification, and optimization (Rasmussen and Williams, 2006). A GP is defined as a stochastic process such that any finite set of random variables has a joint distribution which is the Gaussian distribution. From the definition of a GP, $p(f(x_1), \ldots, f(x_N))$ follows the multivariate Gaussian distribution. Thus, a GP is specified by a mean function $\mu(x_i) = \mathbb{E}f(x_i)$, and a covariance function $k(x_i, x_j) = \mathbb{E}(f(x_i) - \mu(x_i))(f(x_j) - \mu(x_j))^\top$. We express this as $\mathcal{GP}(f|\mu, K)$ where $\mu = (\mu(x_1), \ldots, \mu(x_N))$ and $K_{ij} = k(x_i, x_j)$. We express a GP prior as $p(f|X)$ for simplicity. About the observation noise $p(y|f)$, we simply assume the Gaussian distribution since this is a regression problem. Thus, GP regression is defined as

$$p(y|f) = N(y|f(x), \beta^{-1}I), \tag{2.36}$$

$$p(f|X) = \mathcal{GP}(f|\mu, K), \tag{2.37}$$

where $N(y|f(x), \beta^{-1}I)$ expresses the Gaussian distribution of which mean and variance are $f(x), \beta^{-1}I$ and I denotes the identity matrix and β is a hyperparameter. K is defined by a kernel function and a common choice is

$$k(x_i, x_j) = \lambda_0 e^{-\sum_m \lambda_1^m (x_i^m - x_j^m)^2} + \lambda_2 + \lambda_3 \delta_{i,j}, \tag{2.38}$$

where x_i^m denotes the m-th dimension of the i-th input data and $\{\lambda_i\}_{i=0}^{3}$s are hyperparameters. Based on these notations, our task is to get the posterior distribution $p(f|D) = p(f|X, y)$ and the predictive distribution $p(y_{\text{new}}|x_{\text{new}}, D)$ given new input x_{new}. We can calculate them analytically in GP regression with the Gaussian likelihood function since all the related probability distributions are the Gaussian distributions. The predictive distribution is the Gaussian distribution whose mean

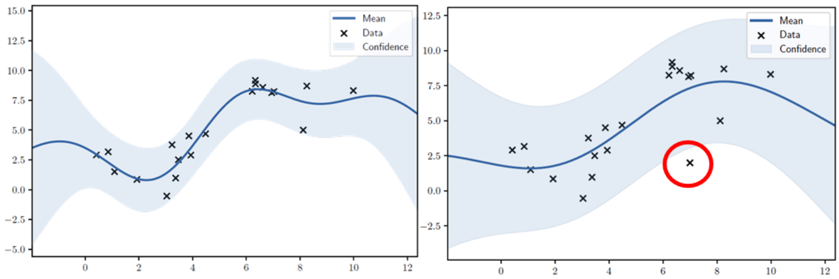


FIGURE 2.3: GP regression using the Gaussian likelihood function

function and variance is given as follows (Bishop, 2006)

$$\begin{cases} \mu(x_{\text{new}}) = k(x_{\text{new}})^\top (K + \beta^{-1} I)^{-1} y, \\ \sigma^2(x_{\text{new}}) = c - k(x_{\text{new}})^\top (K + \beta^{-1} I)^{-1} k(x_{\text{new}}), \end{cases} \tag{2.39}$$

where $k(x_{\text{new}}) = (k(x_1, x_{\text{new}}), \dots, k(x_N, x_{\text{new}})^\top$, $y = (y_1, \dots, y_N)^\top$, and $c = k(x_{\text{new}}, x_{\text{new}}) + \beta^{-1}$. We can predict the output of the new data x_{new} by using the above predictive distribution.

Let us confirm the behavior of GP regression on toy data. Figure.2.3 shows the toy data example; the black crosses show the observed data generated by the relation $y = x + \sin(x) + \epsilon$ where ϵ is generated from the standard Gaussian distribution. In the left figure of Figure.2.3, the blue line shows the mean of the predictive distribution and the shaded area is calculated by the mean plus the variance of the predictive distribution. To check the robustness of GP regression, we artificially added an outlier which is marked by the red circle in the right figure of Figure.2.3. The figure shows that the single outlier has a significant impact on the predictive distribution of GP regression. This means that GP regression is not robust to outliers.

There are various ways to achieve robustness. A standard approach to robustness in statistical inference is a *model-based* method, which uses a log-tail distribution such as the Student-t distribution instead of the Gaussian distribution ((Murphy, 2012)). The density of the k-dimensional Student-t distribution whose mean is μ and the degree of the freedom is v is expressed as

$$\mathrm{St}(x; v, \mu, \Sigma) = \frac{\Gamma((v+k)/2)}{(\pi v)^{k/2} \Gamma(v/2) |\Sigma|^{1/2}} \left(1 + (x-\mu)^\top (v\Sigma)^{-1} (x-\mu)\right)^{-\frac{v+k}{2}}.$$

About its variance, when $v > 2$, it is $\frac{v}{v-2}\Sigma$ and when $1 < v \leq 2$, it diverges.

Fig 2.4 shows the comparison of the Student-t and the Gaussian distribution which have the same means and Σs. In the figure, we can see that the Student-t distribution has a very similar shape to that of the Gaussian distribution, but it has longer tails compared to the Gaussian. The length of the tail is controlled by the degree of the freedom v. When we take the limit of $v \to \infty$, the Student-t distribution is reduced to the Gaussian distribution. The long tail of the Student-t distribution enables us to reduce the effect of outliers. We will clarify the reason for robustness of the Student-t distribution in Section 2.3.3.

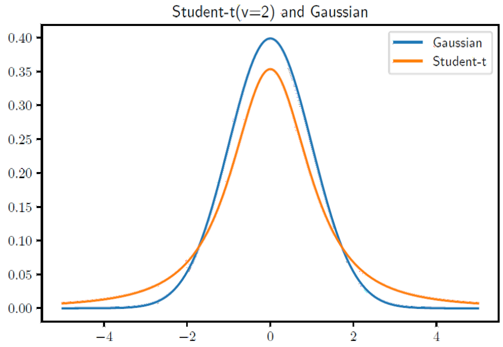


FIGURE 2.4: Comparison of the Student-t and the Gaussian distribution

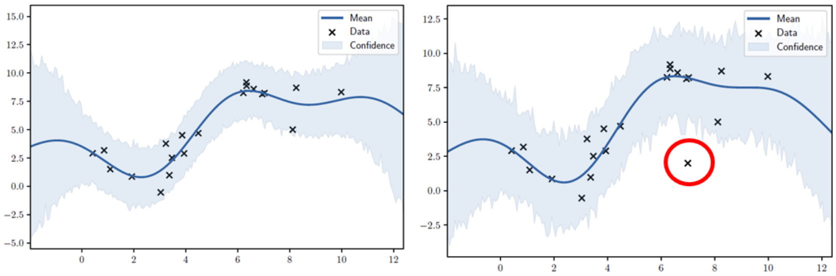


FIGURE 2.5: GP regression using the Student-t likelihood

Let us apply the Student-t distribution to GP regression (Rasmussen and Williams, 2006). We replace the Gaussian likelihood with the Student-t likelihood in GP regression,

$$p(y|f) = \mathrm{St}(y|v, f(x), \beta^{-1}I),$$

where the degree of freedom v is treated as a hyperparameter. With this probabilistic model, we got the predictive distribution shown in Figure.2.5. The left figure is the case that where no outlier exists and the obtained predictive distribution is similar to that of the Gaussian likelihood. In the right figure where an outlier exists, the obtained result is less affected by the outlier compared to that of the Gaussian likelihood. Thus, the Student-t distribution is a promising candidate as the probabilistic model for robust inference. The drawback of using the Student-t distribution is that we cannot get an analytical expression for the posterior and predictive distributions. Thus, we need an approximation method. Due to the long-tail density form, it is known that MC methods suffer from computational inefficiency (Jylänki et al., 2011). The result of Figure.2.5 is obtained by EP approximation which we described in Section 2.2.2.

2.3.3 Influence function (IF)

Here, we discuss why the Gaussian distribution is not robust and the Student-t distribution is robust to outliers by using the *influence function* (IF) (Huber and Ronchetti, 2011). IFs have been used in robust statistics to study how much contamination affects estimated statistics. For a statistic T with an empirical distribution P, the IF at a point z is defined as follows (Huber and Ronchetti, 2011):

$$\begin{aligned}\mathrm{IF}\left(z, T, P\right) &= \frac{\partial}{\partial \varepsilon} T\left(P_{\varepsilon,z}(x)\right)\Big|_{\varepsilon=0} \\ &= \lim_{\varepsilon \to 0} \frac{T\left(P_{\varepsilon,z}(x)\right) - T\left(P(x)\right)}{\varepsilon}. \end{aligned} \tag{2.40}$$

Intuitively, the IF is a relative bias of T caused by contamination at z. Thus, we can measure the robustness of an estimation method with the IF. An important indicator to measure robustness with the IF is

$$\sup_z |\mathrm{IF}\left(z, T, P\right)|.$$

If this indicator diverges, the estimation method is very sensitive to contamination and the effect of outliers can be infinite. If this indicator is bounded, the effect of outliers is bounded and the estimation method is robust to outliers.

Let us check the behavior of the IF of the Gaussian and Student-t distributions. We consider the problem to infer the parameters of the Gaussian and Student-t distributions by ML estimation given the empirical distribution $P_{\varepsilon,z}(x)$, which is defined in Eq.(2.35). Under this setting, we can derive the formula for the IF as

$$\mathrm{IF}\left(z, \theta, P\right) = -\frac{\partial_\theta \ln p(z;\theta)}{\mathbb{E}_P[\partial_\theta \partial_\theta \ln p(x;\theta)]}. \tag{2.41}$$

In the above expression, since the information related to an outlier only appears in the numerator, we can express the IF as (Huber and Ronchetti, 2011)

$$\mathrm{IF}\left(z, \theta, P\right) \propto \partial_\theta \ln p(z;\theta). \tag{2.42}$$

Thus, to study the behavior of the IF, it is sufficient to study the behavior of the score function[2].

With this formula, let us calculate IFs of the Gaussian and Student-t distributions. For simplicity, we only consider the mean parameter and the dimension of the distribution is one. First, we get the score function of the Gaussian distribution as

$$\frac{\partial}{\partial \theta} \ln N(x|\mu,\sigma) \propto \left(x - \mu\right)/\sigma^2. \tag{2.43}$$

Then, we can confirm that

$$\frac{\partial}{\partial \theta} \ln N(x|\mu,\sigma) \xrightarrow[x \to \infty]{} \infty. \tag{2.44}$$

This means that the IF of the Gaussian distribution is not bounded. Thus, the Gaussian distribution is not robust to outliers.

[2] The score function is defined as the gradient of the log likelihood function. Here, the likelihood corresponds to the Gaussian and Student-t distributions.

Next, we consider the Student-t distribution and its score function is

$$\frac{\partial}{\partial\theta}\ln \mathrm{St}(x|v,\mu,\sigma) \propto \frac{(x-\mu)}{\nu\sigma^2+(x-\mu)^2}. \tag{2.45}$$

From this expression, we can confirm the behavior of the IF as

$$\frac{\partial}{\partial\theta}\ln \mathrm{St}(x|v,\mu,\sigma) \xrightarrow[x\to\infty]{} 0. \tag{2.46}$$

This means that the IF of the Student-t distribution is bounded even if an outlier exists at an infinite point. From this expression, we can confirm that the Student-t distribution is robust to outliers and this is a desirable property of the Student-t distribution .

In this way, the IF is a useful tool to analyze the robustness of estimation methods. We will use the IF in Chapter 4.

Chapter 3

Expectation propagation for t-exponential family using q-algebra

In this chapter, we discuss outlier robust inference based on the model-based approach by using long-tail distributions. We present our contribution of the development of an computational efficient algorithm for a generalized exponential family.

3.1 Introduction and summary of this chapter

As we have seen in Chapter 2, the Gaussian distribution is sensitive to outliers and heavier-tailed distributions are preferred in robust inference. For example, the Student-t distribution and a Student-t process (Rasmussen and Williams, 2006; Shah et al., 2014) are good alternatives to the Gaussian distribution (Jylänki et al., 2011) and a Gaussian process (Shah et al., 2014), respectively.

A technical problem of the Student-t distribution is that it does not belong to the exponential family unlike the Gaussian distribution and thus cannot enjoy good properties of the exponential family. To cope with this problem, the exponential family was generalized to the *t-exponential family* (Ding and Vishwanathan, 2010), which contains Student-t distributions as family members. Following this line, the Kullback-Leibler divergence was generalized to the *t-divergence*, and approximation methods based on t-divergence minimization have been explored (Ding et al., 2011). However, the t-exponential family does not allow us to employ standard useful mathematical tricks, e.g., logarithmic transformation does not reduce the product of t-exponential family functions into summation. For this reason, the t-exponential family unfortunately does not inherit an important property of the original exponential family, that is, calculation can be performed through natural parameters. Furthermore, while the dimensionality of sufficient statistics is the same as that of the natural parameters in the exponential family and thus there is no need to increase the parameter size to incorporate new information (Seeger, 2005), this useful property does not hold in the t-exponential family.

The purpose of this chapter is to further explore mathematical properties of natural parameters of the t-exponential family through *pseudo additivity* of distributions based on *q-algebra* used in statistical physics (Nivanen et al., 2003; Suyari and Tsukada, 2005). More specifically, our contributions of this chapter are three-fold:

1. We show that, in the same way as ordinary exponential family distributions, q-algebra allows us to handle the calculation of t-exponential family distributions through natural parameters.

2. Our q-algebra based method enables us to extend *assumed density filtering* (ADF) (Ding et al., 2011) and develop an algorithm of *expectation propagation* (EP) (Minka, 2001) for the t-exponential family. In the same way as the original EP algorithm for ordinary exponential family distributions, our EP algorithm provides a deterministic approximation to the posterior or predictive distribution for t-exponential family distributions with simple moment matching.

3. We apply the proposed EP algorithm to the *Bayes point machine* (Minka, 2001) and *Student-t process classification*, and we demonstrate their usefulness as alternatives to the Gaussian approaches numerically.

3.2 t-exponential family

In this section, we review the *t-exponential family* (Ding and Vishwanathan, 2010; Ding et al., 2011), which is a generalization of the exponential family.

The t-exponential family is defined as,

$$p(x;\lambda) = \exp_t(\langle \Phi(x), \lambda \rangle - g_t(\lambda)), \tag{3.1}$$

where $\exp_t(x)$ is the *deformed exponential function* defined as

$$\exp_t(x) = \begin{cases} \exp(x) & \text{if } t = 1, \\ [1 + (1-t)x]^{\frac{1}{1-t}} & \text{otherwise,} \end{cases} \tag{3.2}$$

and $g_t(\lambda)$ is the log-partition function that satisfies

$$\nabla_\lambda g_t(\lambda) = \mathbb{E}_{p^{\mathrm{es}}}[\Phi(x)]. \tag{3.3}$$

The notation $\mathbb{E}_{p^{\mathrm{es}}}$ denotes the expectation over $p^{\mathrm{es}}(x)$, where $p^{\mathrm{es}}(x)$ is the *escort distribution* of $p(x)$ defined as

$$p^{\mathrm{es}}(x) = \frac{p(x)^t}{\int p(x)^t \mathrm{d}x}. \tag{3.4}$$

We call λ a *natural parameter* and $\Phi(x)$ *sufficient statistics.*

Let us express the k-dimensional Student-t distribution with v degrees of freedom as

$$\mathrm{St}(x;v,\mu,\Sigma) = \frac{\Gamma((v+k)/2)}{(\pi v)^{k/2}\Gamma(v/2)|\Sigma|^{1/2}} \Big(1 + (x-\mu)^\top (v\Sigma)^{-1}(x-\mu)\Big)^{-\frac{v+k}{2}}, \tag{3.5}$$

where $\Gamma(x)$ is the gamma function, $|A|$ is the determinant of matrix A, and $A^\top$ is the transpose of matrix A. We can confirm that the Student-t distribution is a member of the t-exponential family as follows. First, we have

$$\mathrm{St}(x; v, \mu, \Sigma) = \left(\Psi + \Psi \cdot (x-\mu)^\top (v\Sigma)^{-1}(x-\mu)\right)^{\frac{1}{1-t}}, \tag{3.6}$$

$$\text{where } \Psi = \left(\frac{\Gamma((v+k)/2)}{(\pi v)^{k/2}\Gamma(v/2)|\Sigma|^{1/2}}\right)^{1-t}. \tag{3.7}$$

Note that relation $-(v+k)/2 = 1/(1-t)$ holds, from which we have

$$\langle \Phi(x), \lambda \rangle = \left(\frac{\Psi}{1-t}\right)(x^\top K x - 2\mu^\top K x), \tag{3.8}$$

$$g_t(\lambda) = -\left(\frac{\Psi}{1-t}\right)(\mu^\top K \mu + 1) + \frac{1}{1-t}, \tag{3.9}$$

where $K = (v\Sigma)^{-1}$. Then, we can express the Student-t distribution as a member of the t-exponential family as:

$$\mathrm{St}(x; v, \mu, \Sigma) = \left(1 + (1-t)\langle \Phi(x), \lambda \rangle - g_t(\lambda)\right)^{\frac{1}{1-t}} = \exp_t\left(\langle \Phi(x), \lambda \rangle - g_t(\lambda)\right). \tag{3.10}$$

If $t = 1$, the deformed exponential function is reduced to the ordinary exponential function, and therefore the t-exponential family is reduced to the ordinary exponential family, which corresponds to the Student-t distribution with infinite degrees of freedom. For t-exponential family distributions, the *t-divergence* is defined as follows (Ding et al., 2011):

$$D_t(p(x)\|\widetilde{p}(x)) = \int \Big(p^{\mathrm{es}}(x)\ln_t p(x) - p^{\mathrm{es}}(x)\ln_t \widetilde{p}(x)\Big)\mathrm{d}x, \tag{3.11}$$

where

$$\ln_t x := \frac{x^{1-t} - 1}{1-t} \quad (x \geq 0, t \in \mathbb{R}^+) \tag{3.12}$$

and $p^{\mathrm{es}}(x)$ is the escort function of $p(x)$.

3.3 ADF for t-exponential family

We briefly review the assumed density filtering for t-exponential family which was proposed in Ding et al. (2011). This extension is achieved by using the *t-divergence* instead of the KL divergence in the usual ADF in Section 2.2.2:

$$\widetilde{p} = \arg\min_{p'} D_t(p(\theta)\|p'(\theta)) = \arg\min_{p'} \int \Big(p^{\mathrm{es}}(\theta)\ln_t p(\theta) - p^{\mathrm{es}}(\theta)\ln_t p'(\theta;\lambda)\Big)\mathrm{d}\theta. \tag{3.13}$$

When an approximate distribution is chosen from the t-exponential family, we can utilize the property in Ding et al. (2011):

$$\nabla_\lambda g_t(\lambda) = \mathbb{E}_{\widetilde{p^{\mathrm{es}}}}(\Phi(\theta)), \tag{3.14}$$

where $\widetilde{p^{\mathrm{es}}}$ is the escort function of $\widetilde{p}(\theta)$. Then, minimization of the t-divergence yields

$$\mathbb{E}_{p^{\mathrm{es}}}[\Phi(\theta)] = \mathbb{E}_{\widetilde{p^{\mathrm{es}}}}[\Phi(\theta)]. \tag{3.15}$$

This is moment matching, which is a celebrated property of the exponential family. Since the expectation is taken with respect to the escort function, this is called *escort moment matching*.

As an example, let us consider the situation where the prior is the Student-t distribution and the posterior is approximated by the Student-t distribution: $p(\theta|D) \cong \widetilde{p}(\theta) = \mathrm{St}(\theta; \widetilde{\mu}, \widetilde{\Sigma}, v)$. Then the approximated posterior $\widetilde{p}_i(\theta) = \mathrm{St}(\theta; \widetilde{\mu}^{(i)}, \widetilde{\Sigma}^i, v)$ can be obtained by minimizing the t-divergence from $p_i(\theta) \propto \widetilde{p}_{i-1}(\theta)\widetilde{l}_i(\theta)$ as

$$\arg\min_{\mu', \Sigma'} D_t(p_i(\theta) \| \mathrm{St}(\theta; \mu', \Sigma', v)). \tag{3.16}$$

This allows us to obtain an analytical update expression for t-exponential family distributions.

3.4 EP for t-exponential family

As shown in Section 2.2.2, ADF has been extended to EP for an ordinary exponential family (which resulted in moment matching for the exponential family) and ADF is also extended to the t-exponential family (which yielded escort moment matching for the t-exponential family). In this section, we combine these two extensions and propose EP for the t-exponential family.

3.4.1 Pseudo additivity and q-algebra

Differently from ordinary exponential functions, *deformed* exponential functions do not satisfy the product rule:

$$\exp_t(x)\exp_t(y) \neq \exp_t(x+y). \tag{3.17}$$

For this reason, the cavity distribution cannot be computed analytically for the t-exponential family.

On the other hand, the following equality holds for the deformed exponential functions:

$$\exp_t(x)\exp_t(y) = \exp_t(x + y + (1-t)xy), \tag{3.18}$$

which is called *pseudo additivity*.

In statistical physics (Nivanen et al., 2003; Suyari and Tsukada, 2005), a special algebra called *q-algebra* has been developed to handle a system with pseudo additivity. We will use the q-algebra for efficiently handling t-exponential distributions.

Definition 1 (q-product)**.** *Operation* $\otimes_q$ *called the* q-product *is defined as*

$$x \otimes_q y := \begin{cases} [x^{1-q} + y^{1-q} - 1]^{\frac{1}{1-q}} & \text{if } x > 0, y > 0, x^{1-q} + y^{1-q} - 1 > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.19}$$

Definition 2 (q-division)**.** *Operation* $\oslash_q$ *called the* q-division *is defined as*

$$x \oslash_q y := \begin{cases} [x^{1-q} - y^{1-q} - 1]^{\frac{1}{1-q}} & \text{if } \ x > 0, y > 0, x^{1-q} - y^{1-q} - 1 > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.20}$$

Definition 3 (q-logarithm)**.** *The* q-logarithm *is defined as*

$$\ln_q x := \frac{x^{1-q} - 1}{1-q} \quad (x \geq 0, q \in \mathbb{R}^+). \tag{3.21}$$

The q-division is the inverse of the q-product (and visa versa), and the q-logarithm is the inverse of the q-exponential (and visa versa). From the above definitions, the q-logarithm and q-exponential satisfy the following relations:

$$\ln_q(x \otimes_q y) = \ln_q x + \ln_q y, \tag{3.22}$$

$$\exp_q(x) \otimes_q \exp_q(y) = \exp_q(x + y), \tag{3.23}$$

which are called the *q-product rules*. Also for the q-division, similar properties hold:

$$\ln_q(x \oslash_q y) = \ln_q x - \ln_q y, \tag{3.24}$$

$$\exp_q(x) \oslash_q \exp_q(y) = \exp_q(x - y), \tag{3.25}$$

which are called the *q-division rules*.

3.4.2 EP for t-exponential family

The q-algebra allows us to recover many useful properties from the ordinary exponential family. For example, the q-product of t-exponential family distributions yields an unnormalized t-exponential distribution:

$$\exp_t(\langle \Phi(\theta), \lambda_1 \rangle - g_t(\lambda_1)) \otimes_t \exp_t(\langle \Phi(\theta), \lambda_2 \rangle - g_t(\lambda_2)) = \exp_t(\langle \Phi(x), (\lambda_1 + \lambda_2) \rangle - \widetilde{g}_t(\lambda_1, \lambda_2)). \tag{3.26}$$

Based on this q-product rule, we develop EP for the t-exponential family.

Consider the situation where prior distribution $p^{(0)}(\theta)$ is a member of the t-exponential family. As an approximation to the posterior, we choose a t-exponential family distribution

$$\widetilde{p}(\theta; \lambda) = \exp_t(\langle \Phi(\theta), \lambda \rangle - g_t(\lambda)). \tag{3.27}$$

In the original EP for the ordinary exponential family, we considered an approximate posterior of the form

$$\widetilde{p}(\theta) \propto p^{(0)}(\theta) \prod_i \widetilde{l}_i(\theta), \tag{3.28}$$

that is, we factorized the posterior to a product of site approximations corresponding to data. On the other hand, in the case of the t-exponential family, we propose to use the following form called the *t-factorization*:

$$\widetilde{p}(\theta) \propto p^{(0)}(\theta) \otimes_t \prod_i \otimes_t \widetilde{l}_i(\theta). \tag{3.29}$$

The t-factorization is reduced to the original factorization form when $t = 1$.

This t-factorization enables us to calculate EP update rules through natural parameters for the t-exponential family in the same way as the ordinary exponential family. More specifically, consider the case where factor j of the t-factorization is updated in four steps in the same way as original EP.

1. First, we calculate the cavity distribution by using the q-division as

$$\widetilde{p}^{\backslash j}(\theta) \propto \widetilde{p}(\theta) \oslash_t \widetilde{l}_j(\theta) \propto p^{(0)}(\theta) \otimes_t \prod_{i \neq j} \otimes_t \widetilde{l}_i(\theta). \tag{3.30}$$

The above calculation is reduced to subtraction of natural parameters by using the q-algebra rules:

$$\lambda^{\backslash j} = \lambda - \lambda^{(j)}. \tag{3.31}$$

2. The second step is inclusion of site likelihood $l_j(\theta)$, which can be performed by $\widetilde{p}^{\backslash j}(\theta) l_j(\theta)$. The site likelihood $l_j(\theta)$ is incorporated to approximate the posterior by the ordinary product not the q-product. Thus moment matching is performed to obtain a new approximation. For this purpose, the following theorem is useful.

Theorem 1. *The expected sufficient statistic,*

$$\eta = \nabla_\lambda g_t(\lambda) = \mathbb{E}_{\widetilde{p^{\mathrm{es}}}}[\Phi(\theta)], \tag{3.32}$$

can be derived as

$$\eta = \eta^{\backslash j} + \frac{1}{Z_2} \nabla_{\lambda^{\backslash j}} Z_1, \tag{3.33}$$

$$\text{where } \; Z_1 = \int \widetilde{p}^{\backslash j}(\theta)(l_j(\theta))^t \mathrm{d}w, \quad Z_2 = \int \widetilde{p^{\mathrm{es}}}^{\backslash j}(\theta)(l_j(\theta))^t \mathrm{d}w. \tag{3.34}$$

A proof of Theorem 1 is given in Section 3.6.1. After moment matching, we obtain an approximation, $\widetilde{p}_{\mathrm{new}}(\theta)$.

3. Third, we exclude the effect of sites other than j. This is achieved by

$$\widetilde{l}_j^{\mathrm{new}}(\theta) \propto \widetilde{p}_{\mathrm{new}}(\theta) \oslash_t \widetilde{p}^{\setminus j}(\theta), \tag{3.35}$$

which is reduced to subtraction of natural parameter

$$\lambda_{\mathrm{new}}^{\setminus j} = \lambda^{\mathrm{new}} - \lambda^{\setminus j}. \tag{3.36}$$

4. Finally, we update the site approximation by replacing $\widetilde{l}_j(\theta)$ with $\widetilde{l}_j(\theta)^{\mathrm{new}}$.

These four steps are our proposed EP method for the t-exponential family. As we have seen, these steps are reduced to the ordinary EP steps if $t = 1$. Thus, the proposed method can be regarded as an extension of the original EP to the t-exponential family. Since the updates of the proposed EP algorithm is reduced to the simple escort moment matching when the t-exponential family is used for the approximation, the computational cost is the same order as the original EP. This means that our proposed algorithm is also computationally efficient.

3.4.3 Marginal likelihood for t-exponential family

In the above, we omitted the normalization term of the site approximation to simplify the derivation. Here, we derive the marginal likelihood, which requires us to explicitly take into account the normalization term $\widetilde{C}_i$:

$$\widetilde{l}_i(\theta|\widetilde{C}_i, \widetilde{\mu}_i, \widetilde{\sigma}_i^2) = \widetilde{C}_i \otimes_t \exp_t(\langle \Phi(\theta), \lambda \rangle). \tag{3.37}$$

We assume that this normalizer corresponds to Z_1, which is the same assumption as that for the ordinary EP. To calculate Z_1, we use the following theorem (its proof is available in Section 3.6.2):

Theorem 2. *For the Student-t distribution, we have*

$$\int \exp_t(\langle \Phi(\theta), \lambda \rangle - g) \mathrm{d}\theta = \Big(\exp_t(g_t(\lambda)/\Psi - g/\Psi)\Big)^{\frac{3-t}{2}}, \tag{3.38}$$

where g is a constant, $g_t(\lambda)$ is the log-partition function and Ψ is defined in (3.7).

This theorem yields

$$\log_t Z_1^{\frac{2}{3-t}} = g_t(\lambda)/\Psi - g_t^{\setminus j}(\lambda)/\Psi + \log_t \widetilde{C}_j/\Psi, \tag{3.39}$$

and therefore the marginal likelihood can be calculated as follows (see Section 3.6.3 for details):

$$\begin{aligned} Z_{\mathrm{EP}} &= \int p^{(0)}(\theta) \otimes_t \prod_i \otimes_t \widetilde{l}_i(\theta) \mathrm{d}\theta \\ &= \left(\exp_t\Big(\sum_i \log_t \widetilde{C}_i/\Psi + g_t(\lambda)/\Psi - g_t^{\mathrm{prior}}(\lambda)/\Psi\Big)\right)^{\frac{3-t}{2}}. \end{aligned} \tag{3.40}$$

By substituting $\widetilde{C}_i$ in Eq.(3.40), we obtain the marginal likelihood. Note that, if $t = 1$, the above expression of Z_{EP} is reduced to the ordinary marginal likelihood expression (Seeger, 2005). Therefore, this marginal likelihood can be regarded as a generalization of the ordinary exponential family marginal likelihood to the t-exponential family.

In Section 3.6.4 and 3.6.5, we derive specific EP algorithms for the *Bayes point machine* (BPM) and *Student-t process classification*.

3.5 Numerical experiments

In this section, we numerically illustrate the behavior of our proposed EP applied to BPM and Student-t process classification. Suppose that data $(x_1, y_1), \ldots, (x_n, y_n)$ are given, where $y_i \in \{+1, -1\}$ expresses a class label for covariate x_i.

3.5.1 Bayes point machine (BPM)

The likelihood function of BPM is expressed as

$$l_i(\theta) = p(y_i|x_i, \theta) = \epsilon + (1 - 2\epsilon)\Theta(y_i\langle\theta, x_i\rangle), \tag{3.41}$$

where $\Theta(x)$ is the step function taking 1 if $x > 0$ and 0 otherwise and ϵ is the labeling error rate which is treated as a hyperparameter. To estimate the parameter θ, we perform Bayesian inference. We assume that the prior distribution of $\theta$is the Student-t distribution and our task is to obtain the posterior distribution. We cannot obtain the analytical expression of the posterior distribution about this model, we use ADF and EP as approximation methods.

We compare EP and ADF to confirm that EP does not depend on data permutation. We generate a toy dataset in the following way: 1000 data points x are generated from Gaussian mixture model:

$$\begin{aligned} &0.05N(x; [1, 1]^\top, 0.05I) + 0.25N(x; [-1, 1]^\top, 0.05I) \\ &\qquad + 0.45N(x; [-1, -1]^\top, 0.05I) + 0.25N(x; [1, -1]^\top, 0.05I), \end{aligned} \tag{3.42}$$

where $N(x; \mu, \Sigma)$ denotes the Gaussian distribution with respect to x with mean μ and covariance matrix Σ, and I is the identity matrix. For x, we assign label $y = +1$ when x comes from $N(x; [1, 1]^\top, 0.05I)$ or $N(x; [1, -1]^\top, 0.05I)$ and label $y = -1$ when x comes from $N(x; [-1, 1]^\top, 0.05I)$ or $N(x; [-1, -1]^\top, 0.05I)$. We evaluate the dependence of the performance of BPM on data permutation. The derivation of the EP algorithm is shown in Section 3.6.4.

Figure.3.1 shows labeled samples by blue and red points, decision boundaries by black lines which are derived from ADF and EP for the Student-t distribution with $v = 10$ by changing data permutations. The top two graphs show obvious dependence on data permutation by ADF (to clarify the dependence on data permutation, we showed the most different boundary in the figure), while the bottom graph exhibits almost no dependence on data permutations by EP.

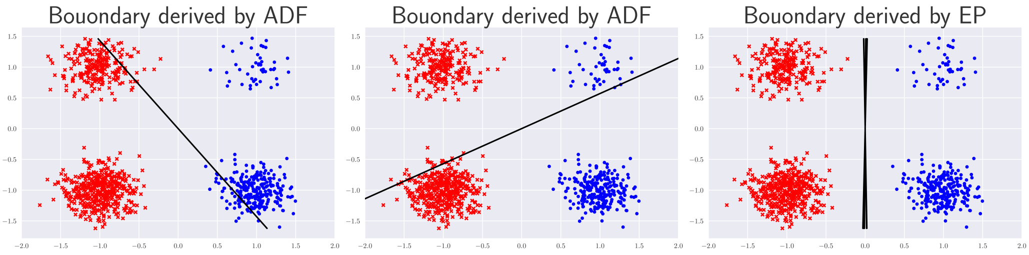


FIGURE 3.1: Boundaries obtained by ADF (left two, with different sample orders) and EP (right).

3.5.2 Student-t process classification

In this section, we propose Student-t process classification (SPC) based on Student-t processes (SPs) (Rasmussen and Williams, 2006; Shah et al., 2014). In Shah et al. (2014), SPs show the superior performance than GPs for the noisy dataset in regression problems. Thus, we compare the performance and the robustness of SPC with Gaussian process classification (GPC).

We first explain a SP. In the case of a GP, the prior distribution $p(f|X)$ is the multivariate Gaussian distribution whose covariance is specified by the kernel function (see Section 2.3.2). In the case of a ST, the prior distribution is the multivariate Student-t distribution which is specified by the covariance kernel and the degree of freedom v. When we use the likelihood $p(y|f)$, we can express the posterior distribution by $p(f|X,y) = \frac{1}{Z}p(f|X)\prod_i p(y_i|f_i)$ where the marginal likelihood is given as $Z = p(y|X) = \int p(f|X)\prod_i p(y_i|f_i)df$ for i.i.d. dataset. Here, we consider a binary classification, and we use the likelihood function:

$$p(y_i|f_i) = l_i(f_i) = \epsilon + (1-2\epsilon)\Theta(y_i f_i). \tag{3.43}$$

This is similar to BPM where the input to the step function is given as a linear model. In GPC and SPC, the input to the step function is a GP and a SP. This GPC model is proposed by Kim and Ghahramani (2008) and this is called robust GPC since the labeling error rate is included in the likelihood function.

Since the posterior distribution cannot be obtained analytically for GPC, we use EP for the ordinary exponential family to approximate the posterior. For the approximation of the posterior distribution of SPC, we use our proposed EP algorithm. The derivation of the EP algorithm is shown in Section 3.6.5.

As numerical experiments, we consider a toy dataset problem and benchmark dataset problems.

We use a two-dimensional toy dataset, where we generate a two-dimensional data point x_i $(i = 1,\ldots,300)$ following the Gaussian distributions: $p(x|y_i = +1) = N(x;[1.5,1.5]^\top, 0.5I)$ and $p(x|y_i = -1) = N(x;[-1,-1]^\top, 0.5I)$. We add eight outliers to the dataset and evaluate the robustness against outliers (about 3% outliers). In the experiment, we used $v = 10$ for Student-t

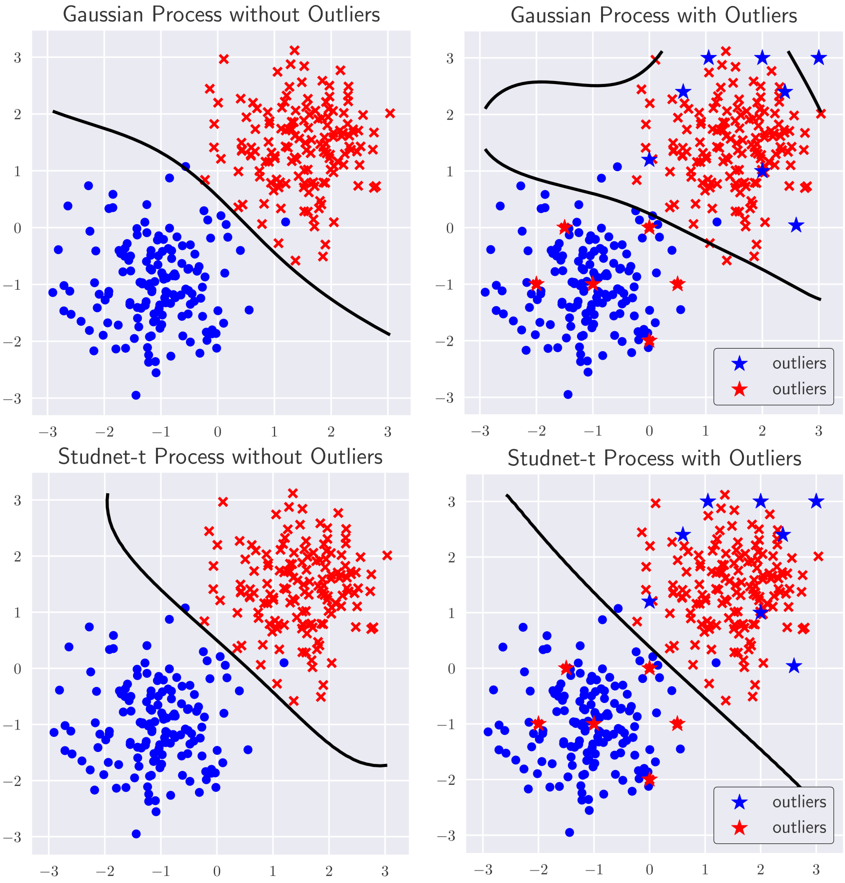


FIGURE 3.2: Classification boundaries.

processes. We furthermore used the following kernel:

$$k(x_i, x_j) = \omega_0 \exp\left\{-\sum_{d=1}^{D} \omega_1^d (x_i^d - x_j^d)^2\right\} + \omega_2 + \omega_3 \delta_{i,j}, \tag{3.44}$$

where x_i^d is the dth element of x_i, and $\omega_0, \omega_1, \omega_2, \omega_3$ are hyperparameters to be optimized. For this optimization, we used the gradient descent algorithm to maximize the marginal log likelihood after the EP algorithm converged. Following the discussion in Hernández-Lobato and Hernández-Lobato (2016), we derive the expression for the gradient of $\log_t Z_{\mathrm{EP}}^{\frac{2}{3-t}}$,

$$\frac{\partial \log_t Z_{\mathrm{EP}}^{\frac{2}{3-t}}}{\partial \omega_j} = \eta^\top \frac{\partial \lambda_{prior}}{\partial \omega_j} - \eta_{prior}^\top \frac{\partial \lambda_{prior}}{\partial \omega_j} + \sum_i \frac{\partial \log_t \widetilde{C}_i}{\partial \omega_j}, \tag{3.45}$$

where, λ_{prior} is the natural parameters of the prior distribution and η_{prior} is the sufficient statistics of the prior distribution. Then we used Adam optimizer for the updates of the algorithm (Kingma and Ba, 2014).

Figure.3.2 shows the labeled samples by blue and red points, the obtained decision boundaries by black lines, and added outliers by blue and red stars. As we can see, the decision boundaries

Table 3.1: Classification Error Rates (%)

Dataset	Outliers	GPC	STC
Pima	0	34.0(3.0)	**32.3(2.6)**
N=767	5%	34.9(3.1)	**32.9(3.1)**
D=8	10%	36.2(3.3)	**34.4(3.5)**
Ionosphere	0	9.6(1.7)	**7.5(2.0)**
N=350	5%	9.9(2.8)	**9.6(3.2)**
D=34	10%	13.0(5.2)	**11.9(5.4)**
Thyroid	0	**4.3(1.3)**	4.4(1.3)
N=207	5%	**4.8(1.8)**	5.5(2.3)
D=5	10%	**5.4(1.4)**	7.2(3.4)
Sonar	0	15.4(3.6)	**15.0(3.2)**
N=207	5%	18.3(4.4)	**17.5(3.3)**
D=60	10%	**19.4(3.8)**	**19.4(3.1)**

Table 3.2: Approximate log evidence

Dataset	Outliers	GPC	STC
Pima	0	-74.1(2.4)	-37.1(6.1)
	5%	-77.8(2.9)	-37.2(6.5)
	10%	-78.6(1.8)	-36.8(6.5)
Ionosphere	0	-59.5(5.2)	-36.9(7.4)
	5%	-75.0(3.6)	-35.8(7.0)
	10%	-90.3(5.2)	-37.4(7.2)
Thyroid	0	-32.5(1.6)	-41.2(4.3)
	5%	-39.1(2.3)	-45.8(5.5)
	10%	-46.9(1.8)	-45.8(4.5)
Sonar	0	-55.8(1.2)	-41.6(1.2)
	5%	-59.4(2.5)	-41.3(1.6)
	10%	-65.8(1.1)	-67.8(2.1)

obtained by the Gaussian process classifier is heavily affected by outliers, while those obtained by the Student-t process classifier are more stable. Thus, as expected, Student-t process classification is more robust against outliers compared to Gaussian process classification, thanks to the heavy-tailed structure of the Student-t distribution.

Next, we compared the performance of Gaussian process and Student-t process classification on the UCI datasets[1]. We used four datasets from the UCI datasets which are widely used for binary classification in GPC experiments. We used cross validation to select the degree of freedom. The range of cross validation for the degree of freedom is from 5 to 15. We used the kernel given in Eq.(3.44). Results are shown in Tables 3.1 and 3.2, where outliers mean how many percentages we randomly flip training dataset labels to make additional outliers. As we can see Student-t process classification outperforms Gaussian process classification in many cases.

3.6 Appendix

In this section, we describe the proofs, supplemental discussion, and detailed explanations for the experimental settings.

3.6.1 Proof of Theorem 1

$$\begin{aligned}
\nabla_{\lambda^{\setminus j}} Z_1 &= \nabla_{\lambda^{\setminus j}} \int \widetilde{p}^{\setminus j}(\theta) l_j(\theta)^t d\theta \\
&= \int (\Psi(\theta) - \nabla_{\lambda^{\setminus j}} g_t(\lambda^{\setminus j})) \widetilde{p^{\mathrm{es}}}^{\setminus j}(\theta) l_j(\theta)^t d\theta \\
&= \int \Psi(\theta) \widetilde{p^{\mathrm{es}}}^{\setminus j}(\theta) l_j(\theta)^t d\theta - \nabla_{\lambda^{\setminus j}} g_t(\lambda^{\setminus j}) \int \widetilde{p^{\mathrm{es}}}^{\setminus j}(\theta) l_j(\theta)^t d.\theta
\end{aligned}$$

[1] `https://archive.ics.uci.edu/ml/index.php`

Using the definition $Z_2 = \int \widetilde{p^{\text{es}}}^{\setminus j}(\theta)(l_j(\theta))^t d\theta$, and $\eta = \nabla_\lambda g_t(\lambda)$,

$$\nabla_{\lambda^{\setminus j}} Z_1 \quad = \quad \eta Z_2 - \eta^{\setminus j} Z_2.$$

Therefore,

$$\eta = \eta^{\setminus j} + \frac{1}{Z_2} \nabla_{\lambda^{\setminus j}} Z_1.$$

3.6.2 Proof of Theorem 2

Here, we consider a one-dimensional distribution. We derive the multivariate formula in the same. Considering the unnormalized t-exponential family, $\exp_t(\langle \Phi(\theta), \lambda \rangle - g)$, and g is a constant, not a true log partition function. We integrate this expression as follows,

$$\begin{aligned}
\int_{-\infty}^{\infty} \exp_t(\langle \Phi(\theta), \lambda \rangle - g) d\theta &= \int_{-\infty}^{\infty} (1 + \Psi(-2\mu^\top K\theta + \theta^\top K\theta) - (1-t)g)^{\frac{1}{1-t}} d\theta \\
&= \int_{-\infty}^{\infty} (1 - \Psi\mu^\top K\mu - (1-t)g + \Psi(\theta - \mu)^\top K(\theta - \mu))^{\frac{1}{1-t}} d\theta \\
&= (1 - \Psi\mu^\top K\mu - (1-t)g)^{\frac{1}{1-t}} \int_{-\infty}^{\infty} \Big(1 + \frac{\Psi(x-\mu)^\top K(x-\mu)}{1 - \Psi\mu^\top K\mu - (1-t)g}\Big)^{\frac{1}{1-t}} d\theta.
\end{aligned}$$

Here, for simplicity, we put $(1 - \Psi\mu^\top K\mu - (1-t)g) = A$, and use the formula, $\int_0^\infty \frac{x^m}{(1+x^2)^n} dx = \frac{1}{2} B\big(\frac{2n-m-1}{2}, \frac{m+1}{2}\big)$, where B denote the beta function. We can get the expression,

$$\int_{-\infty}^{\infty} \exp_t(\langle \Phi(\theta), \lambda \rangle - g) d\theta \quad = \quad \frac{1}{2} B\Big(\frac{3-t}{2(t-1)}, \frac{1}{2}\Big)\Big(\frac{\Psi}{A} K\Big)^{-\frac{1}{2}} A^{\frac{1}{1-t}}.$$

We can proceed with the calculation by using the definition of Ψ, $B(x,y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$, and $\Gamma(\frac{1}{2}) = \sqrt{\pi}$ as follows,

$$\int_{-\infty}^{\infty} \exp_t(\langle \Phi(\theta), \lambda \rangle - g) d\theta \quad = \quad \Psi^{-\left(\frac{1}{2} + \frac{1}{1-t}\right)} A^{\frac{1}{2} + \frac{1}{1-t}}.$$

Here, by using the definition of A and the true log partition function $g_t(\lambda) = \frac{1}{1-t}\big(1 - \Psi(\mu^\top K\mu + 1)\big)$,

$$\begin{aligned}
A^{\frac{1}{2} + \frac{1}{1-t}} &= (1 - \Psi\mu^\top K\mu - (1-t)g)^{\frac{1}{2} + \frac{1}{1-t}} \\
&= (\Psi + (1-t)(g_t(\lambda) - g))^{\frac{1}{2} + \frac{1}{1-t}} \\
&= \Psi^{\frac{1}{2} + \frac{1}{1-t}} (1 + (1-t)(g_t(\lambda) - g)/\Psi)^{\frac{1}{2} + \frac{1}{1-t}}
\end{aligned}$$

Therefore, by substituting this expression into the above integral result, we get the following.

$$\int_{-\infty}^{\infty} \exp_t(\langle \Phi(\theta), \lambda \rangle - g) d\theta = \Big(\exp_t(g_t(\lambda)/\Psi - g/\Psi)\Big)^{\frac{3-t}{2}}$$

3.6.3 Derivation of Eq.(3.40), the marginal likelihood

We calculate the marginal likelihood from the definition:

$$\begin{aligned} Z_{\mathrm{EP}} &= \int p^{(0)}(\theta) \otimes_t \prod_i \otimes_t \widetilde{l}_i(\theta) d\theta \\ &= \int \exp_t\Big(\sum_i \log_t \widetilde{C}_i + \langle \Phi(\theta), \lambda\rangle - g_t^{\mathrm{prior}}(\lambda)\Big) d\theta \\ &= \left(\exp_t\Big(\sum_i \log_t \widetilde{C}_i/\Psi + g_t(\lambda)/\Psi - g_t^{\mathrm{prior}}(\lambda)/\Psi\Big)\right)^{\frac{3-t}{2}}. \end{aligned}$$

3.6.4 Derivation of algorithms for BPM

In this section, we show the details of the update rule of ADF and EP for the Bayes point machine.

3.6.4.1 ADF algorithm for BPM

The detailed update rules of ADF for BPM by t-exponential family are derived in Ding et al. (2011) and given as

$$\begin{aligned} \mu^i &= E_q[\theta] = \mu^{i-1} + \alpha y_i \Sigma^{i-1} x_i, && (3.46) \\ \Sigma^i &= E_q[\theta\theta^\top] - E_q[\theta]E_q[\theta^\top] = r\Sigma^{i-1} - (\Sigma^{i-1}x_i)\left(\frac{\alpha y_i \langle x_i, \mu^i\rangle}{x_i^\top \Sigma^{i-1} x_i}\right)(\Sigma^{i-1}x_i)^\top, && (3.47) \end{aligned}$$

where $\widetilde{q}_i(\theta) \propto \widetilde{p}_i(\theta)^t$, $q_i(\theta) \propto \widetilde{p}_{i-1}(\theta)^t (l_i(\theta))^t$, and

$$\begin{aligned} z &= \frac{y_i \langle x_i, \mu^{i-1}\rangle}{\sqrt{x_i^\top \Sigma^{i-1} x_i}}, && (3.48) \\ Z_1 &= \int \widetilde{p}_{i-1}(\theta)(l_i(\theta))^t d\theta = \epsilon^t + ((1-\epsilon)^t - \epsilon^t) \int_{-\infty}^{z} \mathrm{St}(x; 0, 1, v) dx, && (3.49) \\ Z_2 &= \int \widetilde{q}_{i-1}(\theta)(l_i(\theta))^t d\theta = \epsilon^t + ((1-\epsilon)^t - \epsilon^t) \int_{\infty}^{z} \mathrm{St}(x; 0, v/(v+2), v+2) dx, && (3.50) \\ r &= \frac{Z_1}{Z_2}, && (3.51) \\ \alpha &= \frac{((1-\epsilon)^t - \epsilon^t)\mathrm{St}(z; 0, 1, v)}{Z_2 \sqrt{x_i^\top \Sigma^{i-1} x_i}}. && (3.52) \end{aligned}$$

3.6.4.2 EP algorithm for BPM

For the derivation of the proposed EP algorithm for BPM, we followed the derivation of the ordinary EP algorithm in Minka (2001). We strongly recommend readers to read this article.

Approximate posterior distribution and notations

Natural parameters of Student-t distribution $\mathrm{St}(\theta; v, \mu, \Sigma)$ is $[\lambda_1, \lambda_2]$:

$$\lambda_1 = -2\frac{\Psi K\mu}{1-t}, \tag{3.53}$$

$$\lambda_2 = \frac{\Psi K}{1-t}, \tag{3.54}$$

where $K = (v\Sigma)^{-1}$. From these definitions, we calculate EP update rules through $\Psi K\mu$ and ΨK since natural parameters can be expressed through these two variables.

For BPM, we assume that the whole approximate posterior distribution is expressed k-dimensional $\mathrm{St}(w; m_\theta, V_\theta, v)$ and each site approximation is one-dimensional Student-t like function, $\exp_t(\langle\Phi(\theta), \lambda\rangle)$. This is the function which is the same expression as the Student-t density but unnormalized. Note that for the whole approximation, the degree of freedom is v, but for the site approximation, it is $\tilde{v}$. We describe the relations between them later. As for this site approximation, from Eq.(3.8), we define the i-th site approximation with

$$\begin{aligned}\langle\Phi(\theta), \lambda\rangle &= \frac{\Psi_i}{1-t}\big((\theta^\top x_i)^\top(\tilde{v}\sigma_i)^{-1}(\theta^\top x_i) - 2m_i(\widetilde{v}\sigma_i)^{-1}(\theta^\top x_i)\big)\\ &\propto \frac{\Psi_i}{1-t}\widetilde{v}^{-1}\sigma_i^{-1}(\theta^\top x_i - m_i)^2.\end{aligned} \tag{3.55}$$

Note that the whole posterior approximation is the k-dimensional, but the site approximation is the one-dimensional, therefore the degree of freedom are different from the total approximation and the site approximation to make t consistent. The relation between v, $\widetilde{v}$, and t is given as

$$\frac{1}{t-1} = \frac{v+k}{2} = \frac{\widetilde{v}+1}{2}. \tag{3.56}$$

We express Ψ and K by Ψ_i and K_i for the i-th site approximation. Here, since σ_i is scalar, we can express $K_i = (\widetilde{v}\sigma_i)^{-1}$. If we express $\Psi = (\alpha/|\Sigma|^{1/2})^{1-t}$, then we can express $\Psi_i = (\alpha_i/\sigma_i^{1/2})^{1-t}$ (The definition of α is given in the ADF for BPM). We express Ψ and K by Ψ_θ and K_θ for the whole approximation.

EP update rules

Let us consider the update of the j-th site approximation. Algorithm is composed from following 4 steps.

1. The first step is calculation of cavity distribution, which can be done by

$$\Psi^{\backslash j}K^{\backslash j} = \Psi_\theta(vV_\theta)^{-1} - \Psi_j(\widetilde{v}\sigma_i)^{-1}x_jx_j^\top, \tag{3.57}$$

$$\Psi^{\backslash j}K^{\backslash j}m^{\backslash j} = \Psi_\theta(vV_\theta)^{-1}m_\theta - \Psi_j(\widetilde{v}\sigma_i)^{-1}m_jx_j. \tag{3.58}$$

2. Next step is moment matching. This is calculated in the same way as the ADF update rules. To use the ADF update rule, we have to convert $\Psi^{\backslash j}K^{\backslash j}$ and $\Psi^{\backslash j}K^{\backslash j}m^{\backslash j}$ to $V^{\backslash j}$ and $m^{\backslash j}$,

which are covariance matrix and mean of cavity distribution. When calculating $V^{\backslash j}$ from $\Psi^{\backslash j} K^{\backslash j}$, we have to be careful that $\Psi^{\backslash j}$ contains the determinant of $V^{\backslash j}$. From the definition,

$$\Psi^{\backslash j} K^{\backslash j} = \Big(\frac{\alpha_j}{|V^{\backslash j}|^{1/2}}\Big)^{1-t} (vV^{\backslash j})^{-1}. \tag{3.59}$$

Since α_j and v is the constant, when we put $\frac{V^{\backslash j^{-1}}}{|V^{\backslash j}|^{(1-t)/2}} = B$, following relation holds,

$$|V^{\backslash j}| = \Big(|B|^{\frac{1}{k}}\Big)^{\frac{1}{\frac{t-1}{2} - \frac{1}{k}}}. \tag{3.60}$$

Using this relation, we get $V^{\backslash j}$ and $m^{\backslash j}$.

3. After moment matching, we get V_{new} and m_{new}. Next step is the exclusion step of site other than j. This step is calculated in the same way as the step of cavity distribution.

$$\begin{aligned} \Psi_j K_j &= \Psi_{\text{new}} K_{\text{new}} - \Psi^{\backslash j} K^{\backslash j}, & (3.61) \\ \Psi_j K_j \widetilde{m_j} &= \Psi_{\text{new}} K_{\text{new}} m_{\text{new}} - \Psi^{\backslash j} K^{\backslash j} m^{\backslash j}. & (3.62) \end{aligned}$$

4. To update site parameters, we have to convert $\Psi_j K_j$ and $\Psi_j K_j \widetilde{m_j}$ into σ_j and m_j, which are scalar values. This can be done easily by using the fact that K_j is proportional to $\sigma_j^{-1} x_j x_j^\top$ from the definition.

These steps are the update rules for the site approximation. We have to iterate these steps until site parameters converge.

3.6.5 Derivation of algorithms for Student-t process classification

3.6.5.1 EP algorithm for Student-t process classification

In this section, we show the detailed derivation of the update rules of proposed EP algorithms for Student-t process classification. Since we followed the derivation of the ordinary EP in Hernández-Lobato and Hernández-Lobato (2016); Rasmussen and Williams (2006), we strongly recommend readers to read these articles.

Approximate posterior distribution and notations

Following the ordinary EP algorithm, we approximate the posterior consisting from site approximation. We define the factorizing term that corresponds to data i as follows:

$$\widetilde{l}_i(f_i | \widetilde{C}_i, \widetilde{\mu}_i, \widetilde{\sigma}_i^2) := \widetilde{C}_i \otimes \mathrm{St}(f_i; \widetilde{\mu}_i, \widetilde{\sigma}_i^2, \widetilde{v}). \tag{3.63}$$

For simplicity, we express the unnormalized Student-t like function by $\mathrm{St}(f_i; \widetilde{\mu}_i, \widetilde{\sigma}_i^2, \widetilde{v})$. This is equivalent to $\exp_t(\langle \Phi(f_i), \lambda \rangle)$, where

$$\begin{aligned}\langle \Phi(f_i), \lambda \rangle &= \frac{\Psi_i}{1-t}(f_i^\top K_i f_i - 2\widetilde{\mu}_i^\top K_i f_i) \\ &= \frac{\Psi_i}{1-t}(f_i^\top (v\widetilde{\sigma}_i)^{-1} f_i - 2\widetilde{\mu}_i^\top (v\widetilde{\sigma}_i)^{-1} f_i). \qquad (3.64)\end{aligned}$$

These data corresponding factorizing terms are one-dimensional. In the same way as EP for BPM, the degree of freedom for each site approximation is $\tilde{v}$ and for the whole approximation is v. Note that the whole posterior approximation is the k-dimensional, and site approximation is the one dimensional, the same relation as in the BPM between v, $\widetilde{v}$, and t holds as $\frac{1}{t-1} = \frac{v+k}{2} = \frac{\widetilde{v}+1}{2}$.

The q products of this data corresponding term can be expressed as follows:

$$\prod_i \otimes_t \widetilde{l}_i(f_i) = \mathrm{St}(\widetilde{\mu}, \widetilde{\Sigma}, v) \otimes_t \prod_i \otimes_t \widetilde{C}_i \qquad (3.65)$$

Here, we used the property that q products of Student-t distribution become a Student-t distribution. In the above expression, $\widetilde{\mu}$ is the vector of $\widetilde{\mu}_i$ and $\widetilde{\Sigma}$ is the diagonal and following relations are given,

$$\begin{aligned}\widetilde{K}^{-1} &= (v\widetilde{\Sigma}), & (3.66)\\ \widetilde{\Psi}\widetilde{K} &= \mathrm{diag}(\Psi_1 \mathrm{K}_1 \ldots \Psi_\mathrm{n} \mathrm{K}_\mathrm{n}), & (3.67)\\ & \text{where } \widetilde{\Psi} = \left(\frac{\Gamma((v+k)/2)}{(\pi v)^{k/2}\Gamma(v/2)|\widetilde{\Sigma}|^{1/2}}. \right)^{1-t}. & (3.68)\end{aligned}$$

Therefore, the total form of the approximation of the posterior can be expressed as follows.

$$q(f|X, y) = \mathrm{St}(\mu, \Sigma, v) \propto p(f|X) \otimes_t \Big(\prod_i \otimes_t \widetilde{l}_i(f_i) \Big). \qquad (3.69)$$

From this following relations are obtained,

$$\Psi K = \Psi_0 K_0 + \widetilde{\Psi}\widetilde{K}, \qquad (3.70)$$

$$\Psi K \mu = \widetilde{\Psi}\widetilde{K}\widetilde{\mu}. \qquad (3.71)$$

where $\Psi_0 K_0$ corresponds to the prior distribution, that is $p(f|X)$.

We consider the case that we update site i. For implementation, natural parameter based update rule is preferable. Therefore we define the parameter as follows,

$$\widetilde{\tau}_i = \widetilde{\Psi}_i \widetilde{K}_i, \qquad (3.72)$$

which is the (i,i) element of $\widetilde{\Psi}\widetilde{K}$. We also define,

$$\widetilde{\nu}_i = \widetilde{\Psi}_i \widetilde{K}_i \widetilde{\mu}_i. \qquad (3.73)$$

For the cavity distribution, we define in the same way as,

$$\begin{aligned}\tau_{-i} &= \Psi_{-i}\sigma_{-i}^{-2}\widetilde{v}^{-1}, &(3.74)\\ \nu_{-i} &= \tau_{-i}\mu_{-i}. &(3.75)\end{aligned}$$

EP update rules

1. The first step is to calculate the cavity distribution, we eliminate the effect of site i. To do so, we first integrate out non i terms by using the following formula. Let X and Y are random variable that obey the Student-t distribution,

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathrm{St}\left(\begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}, v\right). \tag{3.76}$$

   The marginal distribution X is given as,

$$X \sim \mathrm{St}\big(\mu_x, \Sigma_{xx}, v\big). \tag{3.77}$$

   By utilizing the above formula, we get

$$\begin{aligned} q_{-i}(f_i) &\propto \int p(f|X) \otimes_t \prod_{j\neq i} \otimes_t l_j(f_j) df_j &(3.78)\\ &\propto \mathrm{St}(\mu_i, \sigma_i^2, v). &(3.79)\end{aligned}$$

   where, μ_i is the ith element of μ and σ_i^2 is the (i,i) element of Σ. In the above expression, the degree of freedom is v in both the joint distribution and marginal distribution. This is unfavorable for our Student-t process. To make the EP procedure consistent with t, we approximate as

$$\begin{aligned} &q_{-i}(f_i) \propto \mathrm{St}(\mu_i, \sigma_i'^2, \widetilde{v}), &(3.80)\\ &\sigma_i'^2 = \sigma_i^2 v/\widetilde{v}. &(3.81)\end{aligned}$$

   This is because for a one-dimensional Student-t distribution, its variance is given by $(v\sigma_i^2)^{-1}$, and in this case, $\widetilde{v} > v$, approximation by $\sigma_i'^2 = \sigma_i^2$ would result in the underestimate of the variance.

   We calculate the cavity distribution in the following way,

$$\begin{aligned}\tau_{-i} &= \widetilde{v}^{-1}{\sigma'}_i^{-2}\Psi_i - \widetilde{\tau}_i, &(3.82)\\ \nu_{-i} &= \widetilde{v}^{-1}{\sigma'}_i^{-2}\Psi_i\mu_i - \widetilde{\nu}_i. &(3.83)\end{aligned}$$

2. Next step is the inclusion of data i to the approximate posterior. This can be done in the same way of BPM. To derive the update rule, we have to convert τ_{-i} and ν_{-i} into σ_{-i}^2 and μ_{-i}. In

this case, the site approximations are one-dimensional, following relation holds,

$$\begin{aligned} \hat{\mu}_i &= \mu_{-i} + \sigma^2_{-i}\alpha, & (3.84) \\ \hat{\sigma^2_i} &= \sigma^2_{-i}(r - \alpha\hat{\mu}_i), & (3.85) \\ \text{where } \alpha = \frac{\big((1-\epsilon)^t - \epsilon^t\big)\mathrm{St}(z :, 0, 1, \widetilde{v})}{Z_2\sqrt{\sigma^2_{-i}}} & \text{ and } z = \frac{y_i\mu_{-i}}{\sqrt{\sigma^2_{-i}}}, & (3.86) \end{aligned}$$

where the definition of Z_2 and r is same as that of BPM. By using σ^2_{-i} and μ_{-i}, we can include the data i information.

3. After the data inclusion step, we exclude the effect other than data i. The calculation of this step can be done in the same way as that of cavity distribution,

$$\begin{aligned} \widetilde{\tau}_i^{\text{new}} &= \widetilde{v}^{-1}\hat{\sigma}_i^{-2}\hat{\Psi}_i - \widetilde{\tau}_{-i}, & (3.87) \\ \widetilde{\nu}_i^{\text{new}} &= \widetilde{v}^{-1}\hat{\sigma}_i^{-2}\hat{\Psi}_i\hat{\mu}_i - \widetilde{\nu}_{-i}. & (3.88) \end{aligned}$$

4. From this $\widetilde{\tau}_i^{\text{new}}$, we can update $\widetilde{\Psi}\widetilde{K}$. Since $\widetilde{\Psi}\widetilde{K}$ is the diagonal matrix, we just update (i, i) element of $\widetilde{\Psi}\widetilde{K}$.

   As a final step, we have to update Σ. To circumvent the calculation of inverse matrix, we put

$$\Delta\tau = -\widetilde{\tau}_i^{\text{new}} - \widetilde{\tau}_{-i} + \widetilde{v}^{-1}\hat{\sigma}_i^{-2}\hat{\Psi}_i. \tag{3.89}$$

   From this, update of ΨK is given as,

$$\Psi^{\text{new}}K^{\text{new}} = \Psi^{\text{old}}K^{\text{old}} + \Delta\tau e_i e_i^\top, \tag{3.90}$$

   where $K^{\text{new}} = (v\Sigma^{\text{new}})^{-1}$ and $K^{\text{old}} = (v\Sigma^{\text{old}})^{-1}$. Here, Σ^{new} is the after the update of Σ and Σ^{old} is the before the update of Σ and e_i is the unit vector of i th direction. By using the matrix formula, that is, for matrix A and B, $(A^{-1} + B^{-1})^{-1} = A - A(A+B)^{-1}A$, we can get the following expression,

$${\Psi^{-1}}^{\text{new}}v\Sigma^{\text{new}} = {\Psi^{-1}}^{\text{old}}v\Sigma^{\text{old}} - \frac{\Delta\tau}{1 + \Delta\tau{\Psi^{-1}}^{\text{old}}v\Sigma^{\text{old}}}s_i s_i^\top, \tag{3.91}$$

   where s_i is the i's column of ${\Psi^{-1}}^{\text{old}}v\Sigma^{\text{old}}$. From ${\Psi^{-1}}^{\text{new}}v\Sigma^{\text{new}}$, we can get Σ^{new}.

These steps are the update rules for the site approximation. We have to iterate these steps until site parameters converge. After EP converges, we optimize the hyperparameters of the kernel function.

3.6.5.2 Prediction rule

Here, we describe how to obtain the prediction for the Student-t process classification. After EP converges, we obtain the expression of the approximate posterior distribution as $q(f|X, y) = \mathrm{St}(\mu, \Sigma, v)$.

When a new point x^* is given, we would like to predict its label y^*. First we calculate the latent variable f^* of x^*. To get the expression of f^*, we use the following lemma (Shah et al., 2014)

Lemma 1. *If $X \sim \mathrm{St}(\mu, \Sigma, v)$, and $x_1 \in R^{n_1}$, $x_2 \in R^{n_2}$ express the first n_1 and remaining n_2 entries of X respectively. Then*

$$x_2|x_1 \sim \mathrm{St}\Big(\widetilde{\mu}_2, \frac{v+\beta_1}{v+n_1} \times \widetilde{\Sigma}_{22}, v+n_1\Big), \tag{3.92}$$

where

$$\widetilde{\mu}_2 = \Sigma_{21}\Sigma_{11}^{-1}(x_1 - \mu_1) + \mu_1, \tag{3.93}$$

$$\widetilde{\Sigma}_{22} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}, \tag{3.94}$$

$$\beta_1 = (x_1 - \mu_1)^\top K_{11}^{-1}(x_1 - \mu_1). \tag{3.95}$$

We consider the following expression,

$$p(\widetilde{f}|X, x^*) = \int p(\widetilde{f}|f, x^*)p(f|X)df. \tag{3.96}$$

The mean of $p(\widetilde{f}|X, x^*)$ is given by

$$\begin{aligned} \mathrm{E}[\widetilde{f}] &= \int \mathrm{E}[p(\widetilde{f}|f, x^*)]p(f|X)df & (3.97)\\ &= \int k^\top \Sigma^{-1} f p(f|X)df & (3.98)\\ &= k^\top \Sigma^{-1}\mu, & (3.99)\end{aligned}$$

where, $k = [k(x^*, x_1), \ldots k(x^*, x_n)]^\top$. Therefore, the prediction of x^* is given by

$$\mathrm{sign}\big(\mathrm{E}[\widetilde{f}]\big) = \mathrm{sign}\big(k^\top \Sigma^{-1}\mu\big). \tag{3.100}$$

Using this expression, we get the decision boundary.

Chapter 4

Variational inference based on robust divergences

In this chapter, we discuss the systematic approach for robust inference by using robust divergences.

4.1 Introduction

In Section 2.3, we discussed the model-based approach to enhance robustness. However, as pointed out in Wang et al. (2017), the model-based method is applicable only to a simple modeling setup.

To handle more complex models, we employ the optimization and variational formulation of Bayesian inference by Zellner (1988). In this formulation, the posterior model is optimized to fit data under the KL divergence, while it is regularized to be close to the prior. In this chapter, we propose the method replacing the KL divergence for data fitting to a robust divergence, such as the β-divergence (Basu et al., 1998) and the γ-divergence (Fujisawa and Eguchi, 2008).

Another robust Bayesian inference method proposed by Ghosh and Basu (2016) follows a similar line to our method, which adopts the β-divergence for pseudo-Bayesian inference. They rigorously analyzed the statistical efficiency and robustness of the method, and numerically illustrated its behavior for the Gaussian distribution.

Our approach can be regarded as an extension of their work to variational inference so that more complex models such as deep networks can be handled. For deep networks with ReLU activation functions, we prove that the *influence function* (IF) (Huber and Ronchetti, 2011) of our proposed inference method is bounded, while it is unbounded in the ordinary variational inference. This implies that our proposed method is robust to both input and output outliers, while the ordinary variational method is not.

In Wang et al. (2017), another robust Bayesian inference method based on a *weighted likelihood* was proposed, where weights are drawn from their prior distribution. They also conducted IF analysis and showed that IF is bounded *asymptotically*. On the other hand, our method is guaranteed to have a bounded IF for finite samples. In addition, by using IF, we numerically show that influence to the predictive distribution by outliers is also bounded in our proposed method.

Finally, we experimentally demonstrate that our robust variational method outperforms ordinary variational inference in regression and classification with neural networks.

4.2 Robust divergences

Here, we review preliminary materials about robust divergences. In Section 2.1.1, we reviewed ML estimation. Given the observed data, we assume the probabilistic model $p(x;\theta)$, ML estimation equation is given by minimizing KL divergence,

$$D_{\mathrm{KL}}\left(\hat{p}(x)\|p(x;\theta)\right) = \mathrm{Const.} - \frac{1}{N}\sum_{i=1}^{N}\ln p(x_i;\theta), \tag{4.1}$$

and this yields

$$0 = \frac{1}{N}\sum_{i=1}^{N}\frac{\partial}{\partial\theta}\ln p(x_i;\theta). \tag{4.2}$$

In terms of robustness, ML estimation is sensitive to outliers because it treats all data points equally. To circumvent this problem, outlier-robust divergence estimation has been developed in statistics. The *density power divergence*, which is also known as the β-divergence, is a vital example (Basu et al., 1998). The β-divergence from functions g to f is defined as

$$D_\beta\left(g\|f\right) = \frac{1}{\beta}\int g(x)^{1+\beta}dx - \frac{\beta+1}{\beta}\int g(x)f(x)^{\beta}dx + \int f(x)^{1+\beta}dx. \tag{4.3}$$

The γ-divergence (Fujisawa and Eguchi, 2008) is another family of robust divergences:

$$D_\gamma\left(g\|f\right) = \frac{1}{\gamma(1+\gamma)}\ln\int g(x)^{1+\gamma}dx - \frac{1}{\gamma}\ln\int g(x)f(x)^{\gamma}dx + \frac{1}{1+\gamma}\ln\int f(x)^{1+\gamma}dx. \tag{4.4}$$

In the limit of $\beta \to 0$ and $\gamma \to 0$, both the β- and γ-divergences converge to the KL divergence:

$$\lim_{\beta\to 0} D_\beta\left(g\|f\right) = \lim_{\gamma\to 0} D_\gamma\left(g\|f\right) = D_{\mathrm{KL}}(g\|f). \tag{4.5}$$

Similarly to ML estimation, minimizing the β-divergence (or the γ-divergence) from empirical distribution $\hat{p}(x)$ to $p(x;\theta)$ gives an empirical estimator:

$$\arg\min_{\theta} D_\beta\left(\hat{p}(x)\|p(x;\theta)\right). \tag{4.6}$$

This yields the following estimating equation:

$$0 = \frac{1}{N}\sum_{i=1}^{N}p(x_i;\theta)^{\beta}\frac{\partial}{\partial\theta}\ln p(x_i;\theta) - \mathbb{E}_{p(x;\theta)}\left[p(x;\theta)^{\beta}\frac{\partial}{\partial\theta}\ln p(x_i;\theta)\right], \tag{4.7}$$

where the second term assures the unbiasedness of the estimator. The first term in Eq.(4.7) is the likelihood weighted according to the power of the probability density for each data point. Since the probability densities of outliers are usually much smaller than those of inliers, those weights effectively suppress the likelihood of outliers.

When $\beta = 0$, all weights become one and thus Eq.(4.7) is reduced to Eq.(4.2). Therefore, adjusting β corresponds to controlling the trade-off between robustness and efficiency.

Eqs.(4.2) and (4.7) are called an M-estimator, and Eq.4.7 is also called a Z-estimator (Huber and Ronchetti, 2011; Basu et al., 1998; Van der Vaart, 1998). In various machine learning applications, those methods showed superior performance (Narayan et al., 2015; Samek et al., 2013; Cichocki et al., 2011).

In Section 4.6.1 and 4.6.2, additional discussions including γ-divergence minimization, supervised settings about robust divergences are given. In Section 4.6.9, we discuss the theoretical difference of β and γ divergence.

4.3 Robust variational inference based on robust divergences

4.3.1 Pseudo posterior distributions

Here, we propose a robust variational inference method based on robust divergences. In Section 2.1.1, we reviewed the reformulation of Bayesian inference as the optimization problem,

$$\mathop{\arg\min}_{q(\theta)\in\mathcal{P}} L(q(\theta)), \tag{4.8}$$

where $\mathcal{P}$ is the set of all probability distributions, $-L(q(\theta))$ is the *evidence lower-bound* (ELBO),

$$L(q(\theta)) = D_{\mathrm{KL}}(q(\theta)\|p(\theta)) - \int q(\theta)\left(-Nd_{\mathrm{KL}}\left(\hat{p}(x)\|p(x|\theta)\right)\right)d\theta, \tag{4.9}$$

and $d_{\mathrm{KL}}\left(\hat{p}(x)\|p(x|\theta)\right)$ denotes the cross-entropy. As detailed in Section 4.6.3, Eq.(4.8) can be equivalently expressed as

$$\mathop{\arg\min}_{q(\theta)\in\mathcal{P}} \mathbb{E}_{q(\theta)}[D_{\mathrm{KL}}\left(\hat{p}(x)\|p(x|\theta)\right)] + \frac{1}{N}D_{\mathrm{KL}}\left(q(\theta)\|p(\theta)\right). \tag{4.10}$$

The first term can be regarded as the expected likelihood (see Eq.(4.1)), while the second term "regularizes" $q(\theta)$ to be close to prior $p(\theta)$.

To enhance robustness to data outliers, let us replace the KL divergence in the expected likelihood term with the β-divergence:

$$\mathop{\arg\min}_{q(\theta)\in\mathcal{P}} \mathbb{E}_{q(\theta)}[D_{\beta}\left(\hat{p}(x)\|p(x|\theta)\right)] + \frac{1}{N}D_{\mathrm{KL}}\left(q(\theta)\|p(\theta)\right). \tag{4.11}$$

Note that Eq.(4.11) can be equivalently expressed as

$$\mathop{\arg\min}_{q(\theta)\in\mathcal{P}} L_{\beta}(q(\theta)), \tag{4.12}$$

where $-L_\beta(q(\theta))$ is the β-ELBO defined as

$$L_\beta(q(\theta) = D_{\mathrm{KL}}(q(\theta)\|p(\theta)) - \int q(\theta)\left(-Nd_\beta\left(\hat{p}(x)\|p(x|\theta)\right)\right)d\theta, \tag{4.13}$$

and $d_\beta(\hat{p}(x)\|p(x|\theta))$ denotes the β-cross-entropy:

$$d_\beta(\hat{p}(x)\|p(x|\theta)) = -\frac{\beta+1}{\beta}\frac{1}{N}\sum_{i=1}^{N} p(x_i|\theta)^\beta + \int p(x|\theta)^{1+\beta}dx.$$

For its solution, we have the following theorem (its proof is available in Section 4.6.4):

Theorem 3. *The solution of Eq.(4.11) is given by*

$$q(\theta) = \frac{e^{-Nd_\beta(\hat{p}(x)\|p(x|\theta))}p(\theta)}{\int e^{-Nd_\beta(\hat{p}(x)\|p(x|\theta))}p(\theta)d\theta}. \tag{4.14}$$

Interestingly, the above expression of $q(\theta)$ is the same as the *pseudo posterior* proposed in Ghosh and Basu (2016).

4.3.2 Discussion about the pseudo posterior distribution

The expression Eq.(4.14) is a member of the pseudo posterior distributions in statistics and it is defined as

$$q(\theta) = \frac{e^{-\lambda R(\theta)}p(\theta)}{\int e^{-\lambda R(\theta)}p(\theta)d\theta}. \tag{4.15}$$

where $p(\theta)$ is prior and $R(\theta)$ expresses the empirical risk, which is not restricted to likelihood and is not necessarily additive in general. This is also called the Gibbs posterior distribution and is extensively studied in the field of PAC Bayesian theory (Germain et al., 2016). The pseudo posterior distribution based on β cross entropy is also expressed as

$$q(\theta) \propto e^{N\left\{\frac{\beta+1}{\beta}\frac{1}{N}\sum_{i=1}^{N} p(x_i;\theta)^\beta + \int p(x;\theta)^{1+\beta}dx\right\}}p(\theta) = \left[\prod_{i}^{N} e^{l_\theta(x_i)}p(\theta)\right], \tag{4.16}$$

where $l_\theta(x_i) = \frac{\beta+1}{\beta}p(x_i;\theta)^\beta - \frac{1}{N}\int p(x;\theta)^{1+\beta}dx$. As discussed in Basu et al. (1998), we can understand the intuitive meaning of Eq.(4.16) by comparing it with the ordinary posterior distribution. In the ordinary posterior distribution, the prior belief is updated by likelihood $p(x_i|\theta)$ which represents the information from data x_i. On the other hand, when using β cross entropy, the prior belief is updated by $e^{l_\theta(x_i)}$ which has information about data x_i. Therefore, although the pseudo posterior is not equivalent to the *posterior distribution* derived by Bayes' theorem, the spirit of updating prior information by observed data is inherited. For this reason, we refer to Eq.(4.14) simply as a *posterior* in this dissertation.

TABLE 4.1: Cross-entropies for robust variational inference.

	Unsupervised	Supervised
β	$-\frac{\beta+1}{\beta}\frac{1}{N}\sum_{i=1}^{N}p(x_i\|\theta)^{\beta}+\int p(x\|\theta)^{1+\beta}dx$	$-\frac{\beta+1}{\beta}\left\{\frac{1}{N}\sum_{i=1}^{N}p(y_i\|x_i,\theta)^{\beta}\right\}+\left\{\frac{1}{N}\sum_{i=1}^{N}\int p(y\|x_i,\theta)^{1+\beta}dy\right\}$
γ	$-\frac{1}{N}\frac{\gamma+1}{\gamma}\sum_{i=1}^{N}\frac{p(x_i\|\theta)^{\gamma}}{\left\{\int p(x\|\theta)^{1+\gamma}dx\right\}^{\frac{\gamma}{1+\gamma}}}$	-$\frac{1}{N}\frac{\gamma+1}{\gamma}\sum_{i=1}^{N}\frac{p(y_i\|x_i,\theta)^{\gamma}}{\left\{\int p(y\|x_i,\theta)^{1+\gamma}dy\right\}^{\frac{\gamma}{1+\gamma}}}$

4.3.3 The pseudo posterior as the solution of the variational problem

The optimization problem (4.11) is generally intractable. Following the same line as the discussion of VI in Section 2.2.1, let us restrict the set of all probability distributions to a set of analytically tractable parametric distributions, $q(\theta;\lambda)\in\mathcal{Q}$. Then the optimization problem yields

$$\underset{q(\theta;\lambda)\in\mathcal{Q}}{\arg\min}\, L_{\beta}(q(\theta;\lambda)).$$

We call this method *β-variational inference* (β-VI).

We optimize objective function L_β by black-box variational inference method and re-parameterization trick (Ranganath et al., 2014). In our implementation, we estimate the gradient of the objective function (4.13) by Monte Carlo sampling.

So far, we focused on the unsupervised learning case and the β-divergence. Actually, we can easily generalize the above discussion to the supervised learning case and also to the γ-divergence, by simply replacing the cross-entropy with a corresponding one shown in Table 4.1. We denote the objective function for the γ-divergence as L_γ in the same way as Eq.(4.13). Note that, there are several choices for the γ-cross-entropy, as detailed in Section 4.6.7. Explicit expression of L, L_β, and L_γ are summarized in Section 4.6.5.

4.4 Influence function analysis

Here, we analyze the robustness of our proposed method based on the *influence function* (IF) (Huber and Ronchetti, 2011). About the definition of IFs, see Section 2.3.3

4.4.1 Derivation of IFs

Now we analyze how posterior distributions derived by VI are affected by contamination. In ordinary VI, we approximate the true posterior with an approximate posterior $q(\theta;\lambda)$ parametrized by λ. The parameter is estimated by maximizing the objective function:

$$L(\lambda) := \mathbb{E}_q \log\left(\frac{p(D|\theta)p(\theta)}{q(\theta;\lambda)}\right). \tag{4.17}$$

TABLE 4.2: Influence functions for robust variational inference.

	Unsupervised	Supervised z=(x’,y’)
$l(z)$	$\ln p\left(z\|\theta\right)$	$\ln p\left(y'\|x',\theta\right)$
$l_\beta(z)$	$\frac{\beta+1}{\beta}p(z\|\theta)^\beta - \int p(x\|\theta)^{1+\beta}dx$	$\frac{\beta+1}{\beta}p(y'\|x',\theta)^\beta - \int p(y\|x',\theta)^{1+\beta}dy$
$l_\gamma(z)$	$\frac{\gamma+1}{\gamma}\frac{p(z\|\theta)^\gamma}{\{\int p(x\|\theta)^{1+\gamma}dx\}^{\frac{\gamma}{1+\gamma}}}$	$\frac{\gamma+1}{\gamma}\frac{p(y'\|x',\theta)^\gamma}{\{\int p(y\|x',\theta)^{1+\gamma}dy\}^{\frac{\gamma}{1+\gamma}}}$

Then, the objective function given by Eq.(4.17) can be regarded as a function of λ whose first-order optimality condition yields

$$0 = \frac{\partial}{\partial\lambda}L\bigg|_{\lambda=\lambda^*}. \tag{4.18}$$

For notational simplicity, we denote $q(\theta;\lambda^*)$ by $q^*(\theta)$.

Recall that the definition of IFs is given as

$$\mathrm{IF}\left(z,T,P\right) = \frac{\partial}{\partial\varepsilon}T\left(P_{\varepsilon,z}(x)\right)\bigg|_{\varepsilon=0} = \lim_{\varepsilon\to0}\frac{T\left(P_{\varepsilon,z}(x)\right) - T\left(P(x)\right)}{\varepsilon}. \tag{4.19}$$

Referring to Eq.(4.19), T corresponds to λ^*, and P is approximated empirically by the training dataset in VI. Then substituting the contaminated distribution:

$$P_{\varepsilon,z}(x) = (1-\varepsilon)P(x) + \varepsilon\Delta_z(x), \tag{4.20}$$

into Eq.(4.17) and using Eq.(4.19) and Eq.(4.18) yield the following theorem (its proof is available in Section 4.6.5):

Theorem 4. *When data contamination is given by Eq.(4.20), IF of ordinary VI is given by*

$$\left(\frac{\partial^2 L}{\partial\lambda^2}\right)^{-1}\frac{\partial}{\partial\lambda}\mathbb{E}_{q^*(\theta)}\left[D_{\mathrm{KL}}(q^*(\theta)\|p(\theta)) + Nl(z)\right], \tag{4.21}$$

IF of β-VI is given by

$$\left(\frac{\partial^2 L_\beta}{\partial\lambda^2}\right)^{-1}\frac{\partial}{\partial\lambda}\mathbb{E}_{q^*(\theta)}\left[D_{\mathrm{KL}}(q^*(\theta)\|p(\theta)) + Nl_\beta(z)\right], \tag{4.22}$$

and IF of γ-VI is given by

$$\left(\frac{\partial^2 L_\gamma}{\partial\lambda^2}\right)^{-1}\frac{\partial}{\partial\lambda}\mathbb{E}_{q^*(\theta)}\left[D_{\mathrm{KL}}(q^*(\theta)\|p(\theta)) + Nl_\gamma(z)\right], \tag{4.23}$$

where $l(z)$, $l_\beta(z)$, and $l_\gamma(z)$ are defined in Table 4.2.

Using these expressions, we analyze how estimated variational parameters can be perturbed by outliers. In practice, it is important to calculate $\sup_z|\mathrm{IF}(z,\theta,P)|$, because if it diverges, the model can be sensitive to small contamination of data.

TABLE 4.3: Behavior of $\sup_z |\mathrm{IF}(z, W, P)|$ in neural networks, "Regression" and "Classification" indicate the cases of ordinary VI, while "β- and γ-Regression" and "β- and γ-Classification" mean that we used β-VI or γ-VI. "Activation function" means the type of activation functions used. "Linear" means that there is no nonlinear transformation, inputs are just multiplied W and added b. $(x_\mathrm{o} : U, y_\mathrm{o} : U)$ means that IF is unbounded while $(x_\mathrm{o} : B, y_\mathrm{o} : U)$ means that IF is bounded for input related outliers, but unbounded for output related outliers.

Activation function	Regression	β- and γ-Regression	Classification	β- and γ-Classification
Linear	$(x_\mathrm{o} : U, y_\mathrm{o} : U)$	$(x_\mathrm{o} : B, y_\mathrm{o} : B)$	$(x_\mathrm{o} : U)$	$(x_\mathrm{o} : B)$
ReLU	$(x_\mathrm{o} : U, y_\mathrm{o} : U)$	$(x_\mathrm{o} : B, y_\mathrm{o} : B)$	$(x_\mathrm{o} : U)$	$(x_\mathrm{o} : B)$
tanh	$(x_\mathrm{o} : B, y_\mathrm{o} : U)$	$(x_\mathrm{o} : B, y_\mathrm{o} : B)$	$(x_\mathrm{o} : B)$	$(x_\mathrm{o} : B)$

4.4.2 IF analysis for specific models

In our analysis, we consider two types of outliers—outliers related to input x and outliers related to output y. For true data generating distributions $p^*(x)$ and $p^*(y|x)$, input-related outlier x_o does not obey $p^*(x)$ and output-related outlier y_o does not obey $p^*(y|x)$. Below we investigate whether IFs are bounded even when $x_\mathrm{o} \to \infty$ or $y_\mathrm{o} \to \infty$.

Although general IF analysis has been extensively carried out in statistics (Huber and Ronchetti, 2011), few works exist focusing on specific models that we often use in recent machine learning applications. Based on this, we consider neural network models for regression and classification (logistic regression). In neural networks, there are parameters $\theta = \{W, b\}$ where outputs of hidden units are calculated by multiplying W to input and then adding b. Our analysis shows that $\sup_z |\mathrm{IF}(z, b, P)|$ is always bounded (see Section 4.6.8 for details), and our exemplary analysis results for $\sup_z |\mathrm{IF}(z, W, P)|$ are summarized in Table 4.3.

From Table 4.3, we can confirm that ordinary VI is always non-robust to output-related outliers. As for input-related outliers, ordinary VI is robust for the "tanh"-activation function, but not for the ReLU and linear activation functions. On the other hand, IFs of our proposed method are bounded for all three activation functions including ReLU. We have further conducted IF analysis for the Student-t likelihood, which is summarized in Section 4.6.8. The notable difference of the behaviors of the influence functions for the Student-t and our proposed VI is that the influence function of our proposed VI converges to 0 as the position of the input related outlier goes to the infinite, on the other hand that of the Student-t likelihood does not converge to 0 but converges to the finite value. This means that there exists some affect from input related outliers even if those outliers are infinitely different from other data. As for the output related outliers, the influence functions of both our method and Student-t likelihood converge to 0 as the output related outliers go to infinite.

Actually, in Bayesian inference, what we really want to know in the end is the *predictive distribution* at test point x_test:

$$p(x_\mathrm{test}|x_{1:N}) = \int p(\theta|x_{1:N}) p(x_\mathrm{test}|\theta) d\theta \approx \int q^*(\theta) p(x_\mathrm{test}|\theta) d\theta.$$

Therefore, it is important to investigate how the predictive distribution is affected by outliers. If the training dataset is contaminated at a rate of ϵ at point z, we can analyze the effect of such data contamination on the predictive distribution by using IFs of the posterior distribution:

$$\frac{\partial}{\partial \epsilon}\mathbb{E}_{q^*(\theta)}\left[p(x_{\text{test}}|\theta)\right] = \frac{\partial \mathbb{E}_{q^*(\theta)}\left[p(x_{\text{test}}|\theta)\right]}{\partial \lambda}\frac{\partial \lambda^*\left(P_{\varepsilon,z}(x)\right)}{\partial \varepsilon}, \tag{4.24}$$

where $\frac{\partial \lambda^*(P_{\varepsilon,z}(x))}{\partial \varepsilon}$ can be analyzed with the IFs derived above. Since analytical discussion on this expression is difficult, we numerically examined its behavior in Section 4.5.2.

The above expression looks similar to the ones derived in Giordano et al. (2015) and Koh and Liang (2017). However, discussion in Giordano et al. (2015) focused on prior perturbation and the formula of IF in Koh and Liang (2017) is applicable only to maximum likelihood estimation and the definition of the contamination is different from ours (see Section 4.6.6 for the details). To our knowledge, ours is the first work to derive IFs of variational inference for data contamination.

4.5 Experiments

Here, we report the experimental results of our proposed method on toy and benchmark datasets. In all the experiments, we used mean-field black-box VI combined with the Adam optimizer (Kingma and Ba, 2014) and assumed that the prior and approximated posterior are both Gaussian.

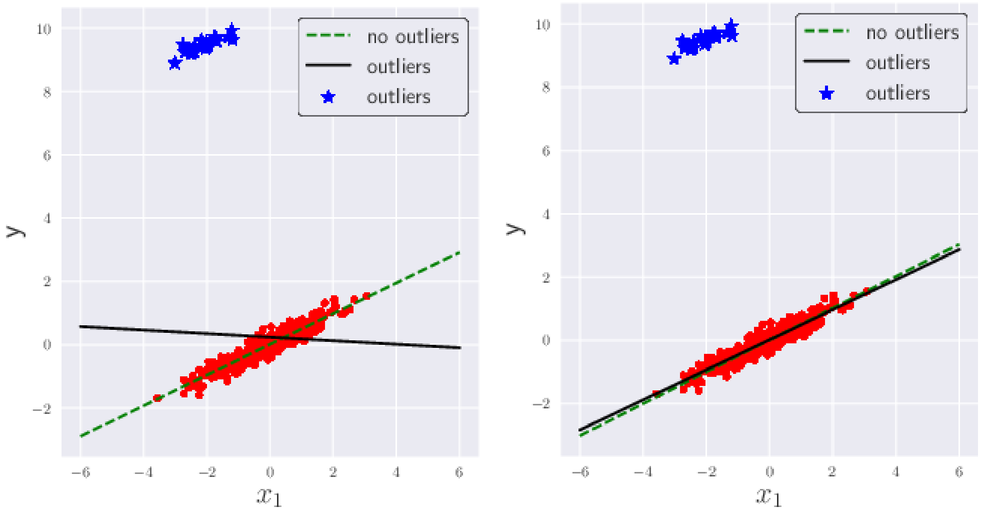
4.5.1 Toy data experiment

We performed a toy dataset experiment for both regression and classification tasks to analyze the performance of the proposed method. We used a two-dimensional toy data and observed how the performance and the predictive distribution are affected by outliers when using ordinary VI and our method. The linear regression and logistic regression models are used.

For the regression task, we generated the toy data by $y \sim w^\top x + \epsilon$, where $x \in \mathbb{R}^2$, $w^\top = (-0.5, -0.1)$, $x \sim N(0, I)$ where I is identity matrix, and $\epsilon \sim N(0, 0.1)$. We generated 1000 data points. Outliers are generated by $x \sim N(-15, 1)$, and we considered them as the measurement error. We generate 24 outliers, which is 2.4$\%$ of the regular dataset. Then we considered the linear regression model, $p(y|x) = N(y|f_\theta(x), 1), f_\theta(x) = Wx + b$.

For the binary classification, the toy data are generated with the probability $p(x|y = +1) = N(x|\mu_1, \sigma_1), p(x|y = -1) = N(x|\mu_2, \sigma_2)$, where $\mu_1^\top = (-1, -1)$, $\mu_2^\top = (1, 1), \sigma_1 = I$, $\sigma_2 = 4I$, where I is identity matrix. We generate 1000 data for each class, and in total 2000 regular points. As outliers we generate 30 outliers by using $p(x|y = +1) = N(x|\mu_o, \sigma_o)$, where $\mu_o^\top = (7, 0)$, $\sigma_2 = 0.1I$. For binary classification, we use logistic regression, where $p(y = +1|x) = logit(f_\theta(x)), f_\theta(x) = Wx + b$.

For regression, the toy data and predictive distribution are shown in Figure. 4.1, where the horizontal axis indicates the first input feature x_1 and the vertical axis indicates the output y. As outliers, we considered input related outliers, which are caused by measurement error. The result of ordinary VI is heavily affected by outliers when there exist outliers, while the result of the proposed method is less affected by outliers.



(a) Ordinary VI with outliers

(b) Proposed VI ($\beta = 0.1$) with outliers

FIGURE 4.1: Linear regression. Predictive distributions are derived by variational inference (VI).

TABLE 4.4: RMSE of VI and β=0.1 VI for toy data.

Outliers	KL(Gaussian)	$\beta = 0.1$(Gaussian)
No outliers	0.01	0.01
Outlier exists	0.69	0.01

For classification, we considered the situation where some of the labels are wrongly specified, as shown in Figure. 5.6. We also illustrated obtained decision boundaries in Figure. 4.2(a), which shows that the ordinary VI based method is heavily affected by outliers and Figure. 4.2(b) shows that our method with $\beta = 0.4$ is less affected by outliers.

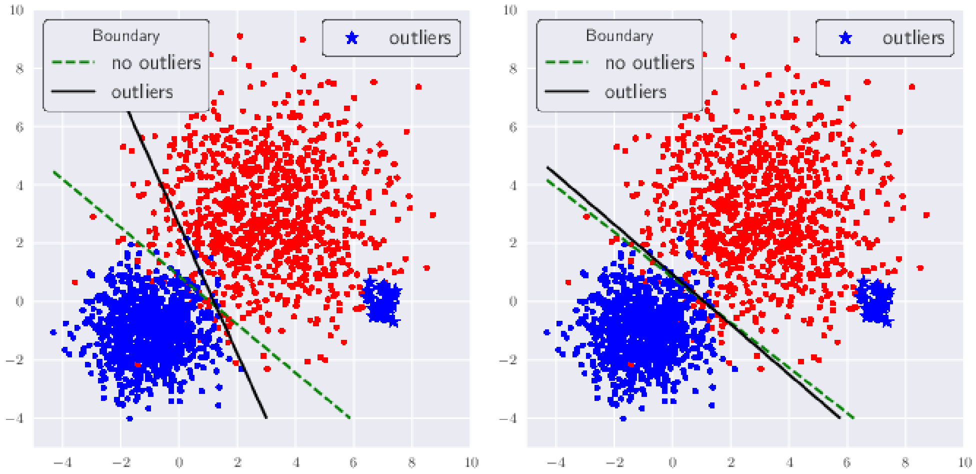
We also show the performance of this toy experiment in Table 4.4 and Table 4.5. Those tables show that the ordinary VI is heavily affected by outliers, while our method is not affected so much. The performance of ordinary VI significantly deteriorates when adding outliers. On the other hand, the performance of our proposing method is not affected by outliers.

4.5.2 Influence to the predictive distribution

Based on Eq.(4.24), we numerically studied the influence of outliers on the predictive distribution. In this study, we used a two-hidden-layer neural network with 20 units in each hidden layer for regression and for classification with logistic loss.

For the calculation of the influence function, we have to evaluate the Hessian of ELBO. To save the computational cost, we used the following relation,

$$\frac{\partial^2 L_\beta}{\partial \lambda^2} v = \arg\min_t \frac{1}{2} t^\top \frac{\partial^2 L_\beta}{\partial \lambda^2} t - v^\top t. \tag{4.25}$$



(a) Ordinary VI with outliers

(b) Proposed VI (β=0.4) with outliers

FIGURE 4.2: Boundaries of logistic regression using ordinary VI and the proposed method

TABLE 4.5: Accuracy of VI and β=0.4 VI for toy data.

Outliers	KL(logistic)	$\beta = 0.4$(logistic)
No outliers	0.97	0.97
Outlier exists	0.95	0.97

This is the technique that instead of calculating the Hessian directly, we calculate the product of the Hessian and a vector by solving the second order optimization problem. In our case, we consider $t = \frac{\partial}{\partial \lambda}\mathbb{E}_{q^*(\theta)}\left[\ln p(x_{\text{test}}|\theta)\right]$ and solve above optimization problem.

Based on the discussion of Section 4.6.8, the dominant term in IF of γ VI behaves similarly as β VI, Thus, we numerically studied the perturbation of predictive distribution for the ordinary VI and β VI. In each calculation, we used 200MC samples to approximate the expectation.

Regression

We used the powerplant dataset in UCI (Lichman, 2013) which has four features for each input. Since it is difficult to visualize the behavior of the influence of predictive distributions, instead, we plot how the log-likelihood of a test point is influenced by an outlier. We compared the influence of ordinary VI based method and proposed method (β=0.1).

We investigated three cases where there are 1) only input related outliers, 2) only output related outliers, and 3) both input and output outliers. In this section, we only show 1) and 2), and the experimental results of 3) are shown in Section 4.6.9.1. To visualize and reduce the computational cost, we contaminated the chosen single feature of the inputs. Since the inputs have 4-dimensional features, $x \in \mathbb{R}^4$, we chose the first feature x_1 to contaminate. To investigate how predictive distribution depends on the contamination of the input, we randomly chose a single data point from the training data and moved the value of the first feature of the chosen training data from $-\infty$ to ∞ as the contamination.

For the output related outlier setting, we randomly chose a single data point from the training data and moved the output value of chosen data from $-\infty$ to ∞. In both input and output related outlier setting, we randomly chose a single data point from the training data and moved the first feature of the input and the output of chosen data from $-\infty$ to ∞ as contamination.

To calculate Eq.(4.24), we have to specify an outlier and a test data point. As an input related outlier, we randomly chose a single data point from the training data and moved the first feature of the chosen data from $-\infty$ to $+\infty$. Similarly, as an output related outlier, we moved randomly chosen output y from $-\infty$ to $+\infty$. As the test data point, we randomly chose a single data point from the test data.

The results are shown in Figure. 4.3, where the horizontal axis indicates the value of the perturbed feature, and the vertical axis indicates the value of $\frac{\partial}{\partial \epsilon}\mathbb{E}_{q^*(\theta)}\left[\ln p(x_{\text{test}}|\theta)\right]$.

The results in Figure. 4.3 show that the model using the ReLU activation inferred by ordinary VI can be affected infinitely by input related outliers, while the influence is bounded in our method. As for output related outliers, models inferred by ordinary VI are infinitely influenced, while influence in our method is bounded. From those results, we can see that our method is robust for both input and output related outliers in the sense that test point prediction is not perturbed infinitely by contaminating a single training point.

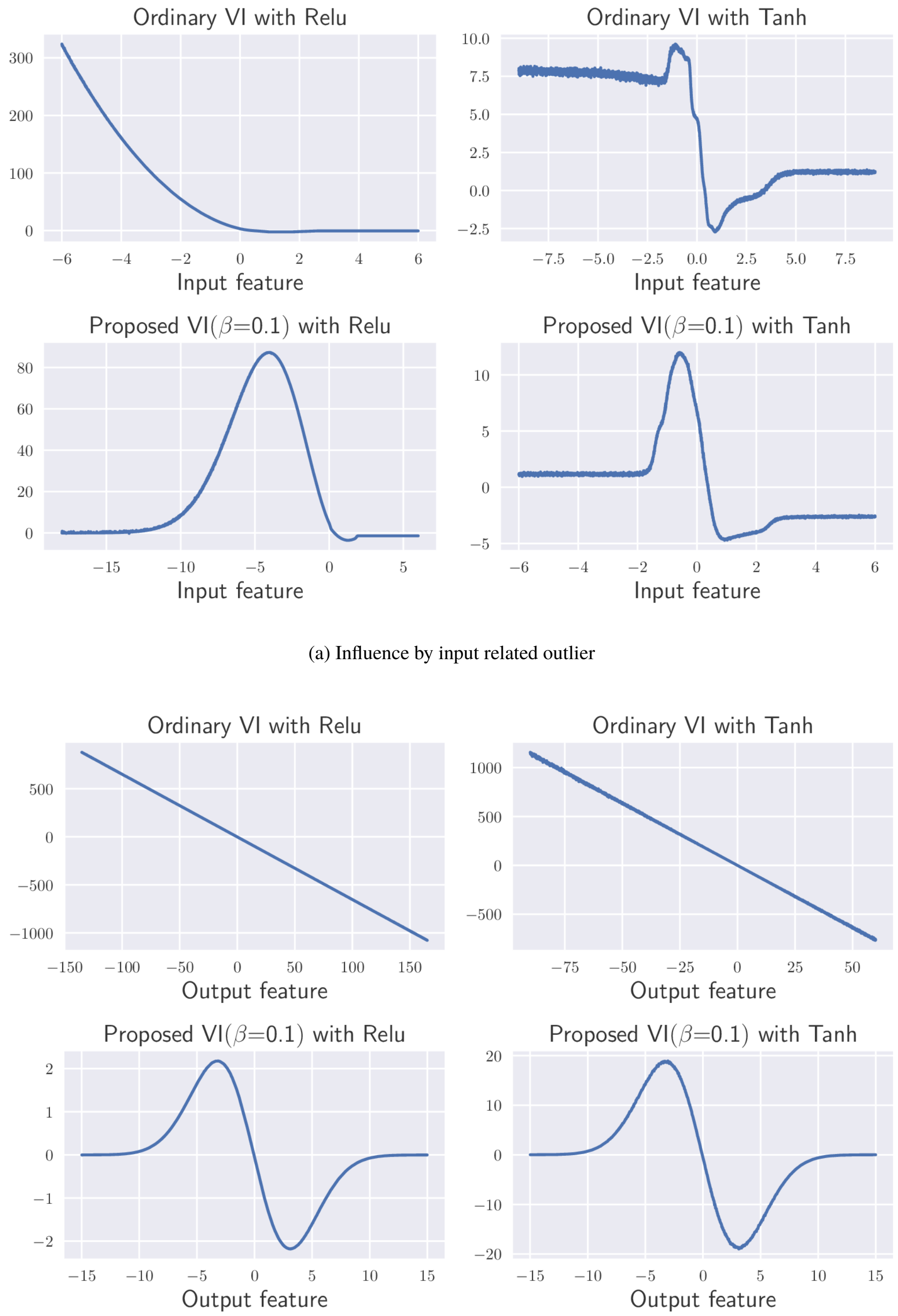
A notable difference from the IF analysis in Section. 4.4.2 is that for the perturbation by input related outliers for the tanh activation function, the value of $\frac{\partial}{\partial \epsilon}\mathbb{E}_{q^*(\theta)}\left[\ln p(x_{\text{test}}|\theta)\right]$, does not converge to zero even for the proposed method in the limit that the absolute value of the input related outlier goes to ∞.

This might be due to the fact that in the limit, the input to the next layer goes to ± 1 when the tanh activation function is used. For the next layer, an input which has value ± 1 might not be so strange compared to regular data, and thus it is not regarded as an outlier. Therefore, during the optimization process, the likelihood of input related outliers is not downweighted so much in the robust divergence and the influence of outliers remains non-zero. If we use the ReLU activation function, in the limit, the input to the next layer becomes much larger than the regular data, and thus it is regarded as an outlier.

Classification

We used the eeg dataset in UCI which has 14 features as input. In the same way as the regression experiment, as an input related outlier, we randomly chose a single data point from the training data and moved the third feature of the chosen data from $-\infty$ to $+\infty$. In the classification problem, first, we considered how predictive distribution depends on the input related outlier. The method is as same as the regression problem. Since inputs have 14-dimensional features, $x \in \mathbb{R}^{14}$, we chose the third feature x_3 to move. The result of how the test log-likelihood is influenced is given in Figure. 4.4. For ordinary VI, using the ReLU activation function causes unbounded influence, while our method keeps the influence bounded. We can also confirm that the influence in our method converges to smaller value than that in ordinary VI in the limit even in the case of tanh.

As an output related outlier, we investigated the influence of label misspecification. We flipped one of the labels in the training data and observed how the test log-likelihood changes. From this



(a) Influence by input related outlier

(b) Influence by output related outlier

FIGURE 4.3: Influence on the test log-likelihood for neural net regression. The horizontal axis indicates the value of the perturbed feature, while the vertical axis indicates the value of $\frac{\partial}{\partial \epsilon}\mathbb{E}_{q^*(\theta)}\left[\ln p(x_{\text{test}}|\theta)\right]$.

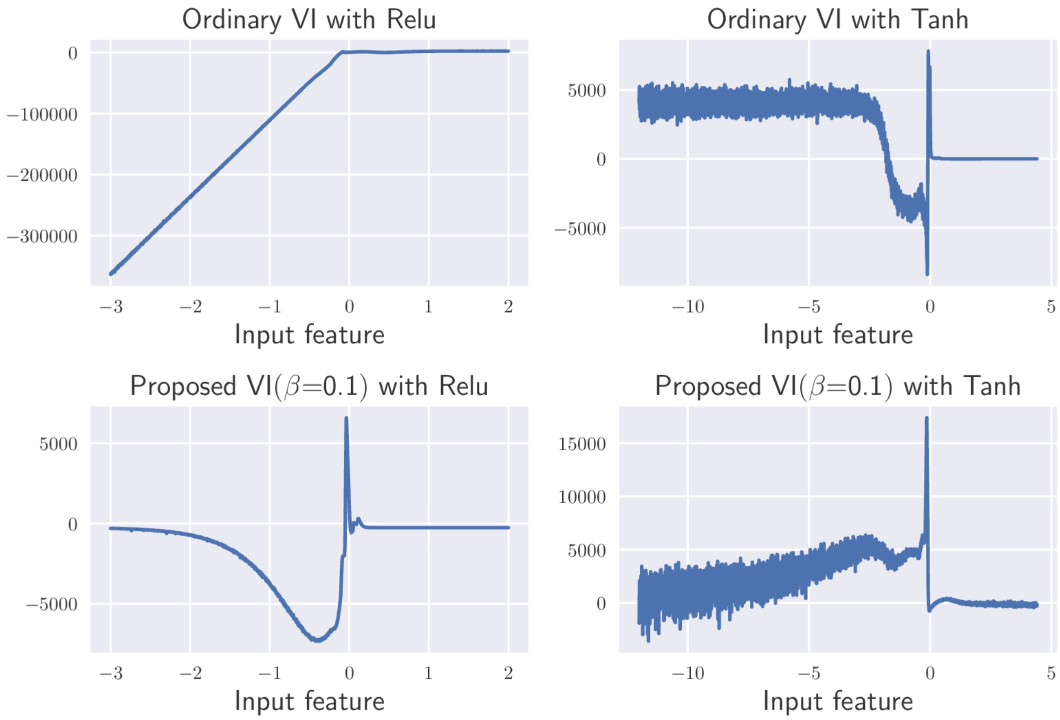


FIGURE 4.4: Influence on the test log-likelihood by input related outlier for neural net classification with logistic loss.

TABLE 4.6: Average change in the test log-likelihood

	Ordinary VI	Proposed VI ($\beta = 0.1$)
ReLU	-1.65e-3	-3.29e-5
tanh	-2.3e-3	-3.49e-4

experiment, we measured how the label misspecification by chosen training data influences the prediction. We repeated this procedure for every training data point and took the average. By this experiment, we measured how one flip of training data would influence the prediction on average. By assuming $\epsilon = \frac{1}{N}$, where N is the number of training data, we calculated

$$\frac{1}{N}\frac{1}{N}\sum_{i}\frac{1}{N_{\text{test}}}\sum_{j}\frac{\partial}{\partial\epsilon_i}\mathbb{E}_{q^*(\theta)}\left[\ln p(y^j_{\text{test}}|x^j_{\text{test}},\theta)\right], \tag{4.26}$$

which represents the averaged amount of change in the test log-likelihood, and the term inside the sum over j means the change in the log-likelihood for the jth test data caused by flipping the label of the ith training data. Without IF, this is difficult to calculate because we have to retrain a neural network with flipped data and this is extremely demanding .

Table 4.6 shows that the change in the test log-likelihood in our method is smaller than that in ordinary VI. This implies that our method is robust against label misspecification.

From these case studies, we confirmed that our method is robust for both input and output related outliers in both regression and classification settings in the sense that the prediction is less influenced

Table 4.7: Test regression accuracy in RMSE

Dataset	Outliers	KL(G)	KL(St)	WL	Rényi	BB-α	β	γ
concrete	0%	8.87(2.57)	**7.34(0.41)**	7.89(0.77)	7.62(0.44)	**7.34(0.31)**	7.58(0.38)	**7.34(0.76)**
N=1030	10%	15.7(2.50)	8.94(2.65)	12.3(2.41)	14.2(1.74)	11.4(2.69)	**8.11(0.89)**	8.26(0.98)
D=8	20%	16.8(0.70)	11.1(3.78)	14.3(2.91)	15.6(1.90)	11.9(2.64)	**8.15(0.99)**	9.25(1.27)
powerplant	0%	4.41(0.13)	4.43(0.15)	4.46(0.17)	4.48(0.15)	4.38(0.83)	**4.37(0.15)**	4.45(0.17)
N=9568	10%	6.44(1.88)	4.54(0.14)	5.12(0.41)	5.49(0.45)	5.91(1.63)	**4.39(0.14)**	4.47(0.16)
D=4	20%	9.97(4.7)	4.56(1.45)	6.44(0.52)	6.87(1.09)	5.52(1.31)	**4.41(0.15)**	4.53(1.46)
protein	0%	5.61(0.38)	**4.79(0.05)**	5.50(0.62)	5.62(0.25)	4.89(0.05)	4.86(0.05)	**4.79(0.04)**
N=45730	10%	6.13(0.02)	4.92(0.05)	6.13(0.03)	6.11(0.03)	6.13(0.03)	4.91(0.04)	**4.90(0.06)**
D=9	20%	6.14(0.03)	4.98(0.07)	6.14(0.03)	6.12(0.03)	6.10(0.28)	4.96(0.05)	**4.95(0.06)**

Table 4.8: Test classification accuracy

Dataset	Outliers	KL	KL(ϵ)	WL	Rényi	BB-α	β	γ
spam	0%	90.9(5.8)	91.2(4.4)	89.2(5.7)	90.0(0.7)	92.9(1.5)	**93.3(1.3)**	92.2(0.8)
N=4601	10%	76.5(37.6)	90.0(5.1)	89.1(5.7)	**92.6(1.4)**	91.6(1.4)	92.4(1.2)	92.1(1.1)
D=57	20%	60.6(48.3)	89.8(5.5)	88.3(5.3)	91.6(1.6)	91.6(1.6)	**92.2(1.3)**	91.6(1.4)
eeg	0%	72.8(2.9)	77.7(3.2)	**81.3(2.4)**	68.4(7.9)	77.5(3.3)	75.9(5.5)	80.2(3.4)
N=14890	10%	56.0(2.6)	62.7(0.09)	56.0(2.4)	57.5(9.6)	67.9(8.2)	60.8(8.1)	**72.5(2.6)**
D=14	20%	56.0(2.7)	60.0(7.1)	56.0(2.4)	57.7(2.4)	67.4(8.8)	56.0(2.4)	**72.2(6.4)**
covertype	0%	65.2(8.8)	73.1(6.2)	**73.4(6.3)**	72.0(6.6)	73.2(4.8)	70.5(5.9)	**73.4(6.1)**
N=581012	10%	60.2(16.9)	**74.4(6.2)**	73.7(5.5)	65.4(8.5)	70.6(5.9)	65.7(9.0)	72.4(7.7)
D=54	20%	56.4(18.7)	71.4(10.4)	71.2(7.2)	67.6(9.7)	67.1(8.1)	66.2(9.6)	**72.3(5.9)**

by outliers.

4.5.3 Choosing β and γ

Finally we show that by choosing parameters β and γ by cross validation, our method can achieve even better performance compared to ordinary VI and other existing robust methods on several benchmark datasets in UCI. In benchmark dataset experiments, we determined β and γ by cross validation and we choose β and γ from 0.1 to 0.9.

We compared proposed methods with several VI methods. KL(G) means ordinary VI with the Gaussian likelihood, KL(St) is ordinary VI with the Student-t likelihood, WL means the method proposed in Wang et al. (2017), Rényi is the Rényi divergence minimization method proposed in Li and Turner (2016) and BB-α is the black-box α divergence minimization method proposed in Hernandez-Lobato et al. (2016) and Li and Gal (2017). For Rényi VI, we chose α from $\{-1.5, -1.0, -0.5, 0.5, 1.0, 1.5\}$ by the cross-validation. For BB-α, we chose α from $\{0, 0.25, 0.5, 0.75, 1.0\}$ by cross-validation. For the Student-t distribution, we chose the degree of freedom from 3 to 10 by cross-validation.

We optimized the variational parameters by black-box variational inference with Adam optimizer and the learning rate at 0.01. For the black-box VI, we use 5 MC samples except for covertype dataset. For the covertype dataset, the learning rate of Adam was set to 0.001 and we used 20 MC samples. The minibatch size of the gradient descent was set to 128.

In both of the regression and classification problems, we artificially increased the percentage of both input and output related outliers in the training dataset. We randomly chose the training dataset for the contamination. To make the input related outliers, we selected the input features for the contamination in the following way. In regression tasks, since the input dimensions of the datasets are not so large, we contaminated all the input features by adding the Gaussian noise. In classification tasks, when the training data has D dimensional features, we randomly chose $D/2$ dimensions to contaminate and add the Gaussian noise. Since the input features had been preprocessed by standardization, the noise we use is the Gaussian distribution which follows $\epsilon \sim N(0, 6I)$ where this magnitude of noise is considered as the kinds of the measurement error. From the numerical calculation of IF, we confirmed that the noise which has "6" times larger variance than the standardization is large enough as outliers. For output related outlier, we randomly chose the dataset for the contamination. In the regression task, we added the Gaussian noise which follows $\epsilon \sim N(0, 6)$ and for the classification task, we flip the label.

Regression

We used a neural net which has two hidden layers each with 20 units and the ReLU activation function. As outliers, we added both input and output related outliers. The experimental results are summarized in Table 5.1. In Table 5.1, "Outliers" means the percentage of outliers in the training dataset we contamined artificially. Our method compares favorably with ordinary VI and existing robust methods for all the datasets. We noticed that when we did not contaminate the datasets, β, γ VI show better performance than other methods. We considered that this is because the training datasets contain some harmful data for the prediction of the test data sets and our proposed method gave small weights to those harmful data. Thus, the predictive performance was improved.

Classification

We used a neural net which has two hidden layers each with 20 units except for the covertype dataset. For the covertype dataset, we used a neural net which has one hidden layer with 50 units. We used the ReLU activation function for all the networks. As outliers, we considered both input and output related outliers. The experimental results are in Table 4.8. In Table 4.8, KL(ϵ) means that we used the robust loss function which is $p(y = 1|g(x, \theta)) = \epsilon + (1 - 2\epsilon)\sigma(g(x, \theta))$, where σ is the sigmoid function, $g(x, \theta)$ is the input to the final layer and ϵ is the hyperparameter.

Our method performs equally to or better than ordinary VI and other existing methods for all the datasets.

4.6 Appendix

In this section, we describe the proofs, supplemental discussion, and detailed explanations for the experimental settings.

4.6.1 Discussion about γ divergence minimization

Usupervised setting

We explain the γ divergence minimization. We minimize the following γ cross entropy,

$$d_\gamma(p^*(x), p(x;\theta)) = -\frac{1}{\gamma}\ln\int p^*(x)p(x;\theta)^\gamma dx + \frac{1}{1+\gamma}\ln\int p(x;\theta)^{1+\gamma}dx. \tag{4.27}$$

This is empirically approximated as

$$L_n(\theta) = d_\gamma(\hat{p}(x), p(x;\theta)) = -\frac{1}{\gamma}\ln\frac{1}{n}\sum_{i=1}^n p(x_i;\theta)^\gamma dx + \frac{1}{1+\gamma}\ln\int p(x;\theta)^{1+\gamma}dx. \tag{4.28}$$

By minimizing $L_n(\theta)$, we can obtain following estimation equation,

$$0 = -\frac{\sum_{i=1}^n p(x_i;\theta)^\gamma \frac{\partial}{\partial\theta}\ln p(x_i;\theta)}{\sum_{i=1}^n p(x_i;\theta)^\gamma} + \int \frac{p(x;\theta)^{1+\gamma}}{\int p(x;\theta)^{1+\gamma}dx}\frac{\partial}{\partial\theta}\ln p(x;\theta)dx. \tag{4.29}$$

This is a weighted likelihood equation, where the weights are $\frac{p(x_i;\theta)^\gamma}{\sum_{i=1}^n p(x_i;\theta)^\gamma}$. The second term is for the unbiasedness of the estimating equation.

In Eq.(4.29), we derived the gamma divergence minimization equation for the unsupervised setting. This estimation equation is equivalent to minimizing following expression,

$$L'_n(\theta) = -\frac{1}{n}\sum_{i=1}^n \frac{\gamma+1}{\gamma}\frac{p(x_i|\theta)^\gamma}{\left\{\int p(x|\theta)^{1+\gamma}dy\right\}^{\frac{\gamma}{1+\gamma}}}. \tag{4.30}$$

In the following derivation, we use $L'_n(\theta)$ as γ cross entropy instead of using the original form of γ cross entropy. The reason is discussed in Section 4.6.7.

Supervised setting

Here, we explain the γ divergence minimization for the supervised setting. We denote the true distribution as $p^*(y,x) = p^*(y|x)p^*(x)$. We denote the regression model by $p(y|x;\theta)$.

Following Fujisawa and Eguchi (2008), we define the divergence between true distribution and the model by

$$\begin{aligned} &D_\gamma(p^*(y|x), p(y|x;\theta); p^*(x)) \\ &= \frac{1}{1+\gamma}\ln\int\left\{\int p(y|x;\theta)^{1+\gamma}dy\right\}p^*(x)dx - \frac{1}{\gamma}\ln\int\left\{\int p^*(y|x)p(y|x;\theta)^\gamma dy\right\}p^*(x)dx + \text{Const.}\end{aligned} \tag{4.31}$$

As discussed in Fujisawa and Eguchi (2008), in the limit where $\gamma \to 0$, this divergence becomes ordinary KL divergence,

$$\lim_{\gamma\to 0} D_\gamma(p^*(y|x), p(y|x;\theta)|p^*(x)) = \int D_{\mathrm{KL}}(p^*(y|x), p(y|x;\theta))p^*(x)dx. \tag{4.32}$$

What we minimize is following γ cross entropy over the distribution $p^*(x)$. Actually, minimizing γ divergence is equivalent to minimizing the second term of Eq.(4.31). By empirical approximation, what we minimize is following expression,

$$L_n(\theta) = -\frac{1}{n}\sum_{i=1}^{n} l_n(\theta) = -\frac{1}{n}\sum_{i=1}^{n} \frac{p(y_i|x_i;\theta)^{\gamma}}{\left\{\int p(y|x_i;\theta)^{1+\gamma}dy\right\}^{\frac{\gamma}{1+\gamma}}}. \tag{4.33}$$

As $\gamma \to 0$, above expression goes to

$$L_n(\theta) = -\frac{1}{n}\sum_{i=1}^{n} \ln p(y_i|x_i;\theta). \tag{4.34}$$

This is ordinary KL cross entropy.

4.6.2 Discussion about β divergence minimization

Here, we consider supervised setup for β divergence minimization. The empirical approximation of β cross entropy for supervised settings is

$$L_n(\theta) = d_\beta(\hat{p}(y|x), p(y|x;\theta); \hat{p}(x)) = -\frac{\beta+1}{\beta}\left\{\frac{1}{n}\sum_{i=1}^{n} p(y_i|x_i;\theta)^{\beta}\right\} + \left\{\frac{1}{n}\sum_{i=1}^{n}\int p(y|x_i;\theta)^{1+\beta}dy\right\}. \tag{4.35}$$

For the unsupervised setting, the empirical approximation of β cross entropy is

$$L_n(\theta) = d_\beta(\hat{p}(x), p(x;\theta)) = -\frac{\beta+1}{\beta}\frac{1}{n}\sum_{i=1}^{n} p(x_i;\theta)^{\beta} + \int p(x;\theta)^{1+\beta}dx. \tag{4.36}$$

4.6.3 Proof of Eq.(4.10)

From the definition of KL divergence Eq.(4.1), the cross entropy can be expressed as

$$d_{\mathrm{KL}}\left(\hat{p}(x)\|p(x|\theta)\right) = D_{\mathrm{KL}}\left(\hat{p}(x)\|p(x|\theta)\right) + \mathrm{Const.} \tag{4.37}$$

By substituting the above expression into the definition of $L(q(\theta))$, we obtain

$$L(q(\theta)) = D_{\mathrm{KL}}(q(\theta)\|p(\theta)) + N\mathbb{E}_{q(\theta)}[D_{\mathrm{KL}}\left(\hat{p}(x)\|p(x|\theta)\right)] + \mathrm{Const.}$$

What we have to consider is

$$\operatorname*{arg\,min}_{q(\theta)\in\mathcal{P}} L(q(\theta)), \tag{4.38}$$

We can disregard the constant term in $L(q(\theta))$, and above optimization problem is equivalent to

$$\operatorname*{arg\,min}_{q(\theta)\in\mathcal{P}} \frac{1}{N} L(q(\theta)). \tag{4.39}$$

Therefore Eq.(4.10) is equivalent to Eq.(4.17)

4.6.4 Proof of Theorem 3

The objective function is given as

$$L_\beta = \mathbb{E}_{q(\theta)}[D_\beta\left(\hat{p}(x)||p(x|\theta)\right)] + \lambda' D_{KL}\left(q(\theta)||p(\theta)\right), \tag{4.40}$$

where λ' is the regularization constant. We optimize this with the constraint that $\int q(\theta)d\theta = 1$. We calculate using the method of variations and Lagrange multipliers, we can get the optimal $q(\theta)$ in the following way,

$$\frac{d(L_\beta + \lambda(\int q(\theta)d\theta - 1))}{dq(\theta)} = D_\beta\left(\hat{p}(x)|p(x|\theta))\right] + \lambda' \ln\frac{q(\theta)}{p(\theta)} - (1+\lambda) = 0. \tag{4.41}$$

By rearranging the above expression, we can get the following relation,

$$q(\theta) \propto p(\theta)e^{-\frac{1}{\lambda'}d_\beta(\hat{p}(x)|p(x|\theta))} \tag{4.42}$$

If we set $\frac{1}{\lambda'} = N$ and normalize the above expression, we get the Theorem 3, that is,

$$q(\theta) = \frac{e^{-Nd_\beta(\hat{p}(x)|p(x|\theta))}p(\theta)}{\int e^{-Nd_\beta(\hat{p}(x)|p(x|\theta))}p(\theta)d\theta}. \tag{4.43}$$

We can get the similar expression for γ cross entropy.

Interestingly, if we use KL cross entropy instead of β cross entropy in the above discussion, following relation holds,

$$\begin{aligned} q(\theta) \propto p(\theta)e^{-\frac{1}{\lambda'}d_{KL}(\hat{p}(x)|p(x|\theta))} &= p(\theta)e^{-N(-\frac{1}{N}\sum_i \ln p(x_i|\theta))} \\ &= p(\theta)\prod_i p(x_i|\theta) \\ &= p(\theta)p(D|\theta). \end{aligned} \tag{4.44}$$

The normalizing constant is

$$\int p(\theta)\prod_i p(x_i|\theta)d\theta = p(D). \tag{4.45}$$

Finally, we get the optimal $q(\theta)$

$$q(\theta) = \frac{p(D|\theta)p(\theta)}{p(D)}. \tag{4.46}$$

This is the posterior distribution which can be derived by Bayes' theorem.

In the above proof, we set regularization constant as $\frac{1}{\lambda'} = N$ to derive the expression. In this dissertation, we only consider the situation that regularization constant is $\frac{1}{\lambda'} = N$ based on the

TABLE 4.9: Cross-entropies for robust variational inference.

	Unsupervised	Supervised
d_β	$-\frac{\beta+1}{\beta}\frac{1}{N}\sum_{i=1}^N p(x_i\|\theta)^\beta + \int p(x\|\theta)^{1+\beta}dx$	$-\frac{\beta+1}{\beta}\left\{\frac{1}{N}\sum_{i=1}^N p(y_i\|x_i,\theta)^\beta\right\} + \left\{\frac{1}{N}\sum_{i=1}^N \int p(y\|x_i,\theta)^{1+\beta}dy\right\}$
d_γ	$-\frac{1}{N}\frac{\gamma+1}{\gamma}\sum_{i=1}^N \frac{p(x_i\|\theta)^\gamma}{\left\{\int p(x\|\theta)^{1+\gamma}dx\right\}^{\frac{\gamma}{1+\gamma}}}$	-$\frac{1}{N}\frac{\gamma+1}{\gamma}\sum_{i=1}^N \frac{p(y_i\|x_i,\theta)^\gamma}{\left\{\int p(y\|x_i,\theta)^{1+\gamma}dy\right\}^{\frac{\gamma}{1+\gamma}}}$

similarity of Bayes' theorem.

4.6.5 Proof of Theorem 4

We consider the situation where the distribution is expressed as

$$P_{\varepsilon,z}\left(x\right) = \left(1-\varepsilon\right)P_n\left(x\right) + \varepsilon\Delta_z\left(x\right). \tag{4.47}$$

Before going to the detail, we summarize the objective function of VI and proposed method. First, the objective function of ordinary VI is given by

$$L = D_{\mathrm{KL}}(q(\theta)\|p(\theta)) + N\mathbb{E}_{q(\theta)}\left[Nd_{\mathrm{KL}}\left(\hat{p}(x)\|p(x|\theta)\right)\right]. \tag{4.48}$$

In the same way, objective functions of β-VI and γ-VI are given by

$$L_\beta = D_{\mathrm{KL}}(q(\theta)\|p(\theta)) + N\mathbb{E}_{q(\theta)}\left[Nd_\beta\left(\hat{p}(x)\|p(x|\theta)\right)\right], \tag{4.49}$$

$$L_\gamma = D_{\mathrm{KL}}(q(\theta)\|p(\theta)) + N\mathbb{E}_{q(\theta)}\left[Nd_\gamma\left(\hat{p}(x)\|p(x|\theta)\right)\right], \tag{4.50}$$

where d_β and d_γ are summarized in Table 4.9. By using these expressions, we will derive the IFs.

4.6.5.1 Derivation of the IF for ordinary VI

We start from the first order condition,

$$\begin{aligned} 0 &= \frac{\partial}{\partial\lambda}L\bigg|_{\lambda=\lambda^*} \\ &= \nabla_\lambda\mathbb{E}_{q(\theta;\lambda^*(\epsilon))}\left[N\int dP_{\epsilon,z}(x)\ln p(x|\theta) + \ln p(\theta) - \ln q(\theta;\lambda^*(\epsilon))\right]. \end{aligned} \tag{4.51}$$

We differentiate above expression with respect to ϵ, then we obtain following expression,

$$\begin{aligned} 0 = \nabla_\lambda\int d\theta\frac{\partial\lambda^*(\epsilon)}{\partial\epsilon}\frac{\partial q}{\partial\lambda^*(\epsilon)}&\left\{(1-\epsilon)N\int dP_n(x)\ln p(x|\theta) + \epsilon N\ln p(z|\theta) + \ln p(\theta)\right\} \\ + \nabla_\lambda\mathbb{E}_{q(\theta;\lambda^*(\epsilon))}&\left[-N\int dP_n(x)\ln p(x|\theta) + N\ln p(z|\theta)\right] \\ - \nabla_\lambda\int d\theta\frac{\partial\lambda^*(\epsilon)}{\partial\epsilon}\frac{\partial q}{\partial\lambda^*(\epsilon)}&\ln q(\theta;\lambda^*(\epsilon)) - \nabla_\lambda\mathbb{E}_{q(\theta;\lambda^*(\epsilon))}\left[\frac{\partial\lambda^*(\epsilon)}{\partial\epsilon}\cdot\frac{\partial\ln q}{\partial\lambda^*(\epsilon)}\right]. \end{aligned} \tag{4.52}$$

From above expression, if we take $\epsilon \to 0$, we soon obtain following expression,

$$\frac{\partial \lambda^* (\varepsilon)}{\partial \varepsilon} = -\left(\frac{\partial^2 L}{\partial \lambda^2}\right)^{-1} \frac{\partial}{\partial \lambda} \mathbb{E}_{q(\theta)} \left[N \int dP_n(x) \ln p\left(x|\theta\right) - N \ln p\left(z|\theta\right) \right]. \tag{4.53}$$

Actually, this can be transformed to following expression by using the first order condition,

$$\frac{\partial \lambda^* (\varepsilon)}{\partial \varepsilon} = \left(\frac{\partial^2 L}{\partial \lambda^2}\right)^{-1} \frac{\partial}{\partial \lambda} \mathbb{E}_{q(\theta)} \left[D_{KL}(q(\theta;\lambda)|p(\theta)) + N \ln p\left(z|\theta\right) \right]. \tag{4.54}$$

4.6.5.2 Derivation of the IF for β VI

Next we consider IF for β VI. To proceed calculation, we have to be careful that empirical approximation of β cross entropy takes different form between unsupervised and supervised setting as shown in Eq.(4.36) and Eq.(4.35).

For the unsupervised situation, we can write the first order condition as,

$$\begin{aligned} 0 &= \frac{\partial}{\partial \lambda} L_\beta \bigg|_{\lambda=\lambda^*} \\ &= \nabla_\lambda \mathbb{E}_{q(\theta;\lambda^*(\epsilon))} \left[N \int dP_{\epsilon,z}(x) \frac{\beta+1}{\beta} p(x|\theta)^\beta - N \int p(x|\theta)^{1+\beta} dx + \ln p(\theta) - \ln q(\theta;\lambda^*(\epsilon)) \right]. \end{aligned} \tag{4.55}$$

We can proceed calculation in the same way as ordinary VI. We get the following expression

$$\frac{\partial \lambda^* (\varepsilon)}{\partial \varepsilon} = -\frac{\beta+1}{\beta} \left(\frac{\partial^2 L_\beta}{\partial \lambda^2}\right)^{-1} \frac{\partial}{\partial \lambda} \mathbb{E}_{q(\theta)} \left[N \int dP_n(x) p(x|\theta)^\beta - N p\left(z|\theta\right)^\beta \right]. \tag{4.56}$$

Next, we consider the supervised situation. We consider the situation where the contamination is expressed as

$$P_{\varepsilon,z=(x',y')}\left(x,y\right) = \left(1-\varepsilon\right) P_n\left(x,y\right) + \varepsilon \Delta_{z=(x',y')}\left(x,y\right). \tag{4.57}$$

The first order condition is

$$\begin{aligned} 0 &= \frac{\partial}{\partial \lambda} L_\beta \bigg|_{\lambda=\lambda^*} \\ &= \nabla_\lambda \mathbb{E}_{q(\theta;\lambda^*(\epsilon))} \left[N \int dP_{\epsilon,z}(x,y) \frac{\beta+1}{\beta} p(y|x,\theta)^\beta - N \int dP_{\epsilon,x'}(x) \left\{ \int p(y|x,\theta)^{1+\beta} dy \right\} \right] \\ &+ \nabla_\lambda \mathbb{E}_{q(\theta;\lambda^*(\epsilon))} \left[\ln p(\theta) - \ln q(\theta;\lambda^*(\epsilon)) \right]. \end{aligned} \tag{4.58}$$

We can proceed the calculation and derive the IF as follows

$$\begin{aligned}\frac{\partial \lambda^*(\varepsilon)}{\partial \varepsilon} =& -N\left(\frac{\partial^2 L_\beta}{\partial \lambda^2}\right)^{-1}\frac{\partial}{\partial \lambda}\mathbb{E}_{q(\theta)}\left[\frac{\beta+1}{\beta}\left(\int dP_n(y,x)p(y|x,\theta)^\beta - p\left(y'|x',\theta\right)^\beta\right)\right]\\ &+N\left(\frac{\partial^2 L_\beta}{\partial \lambda^2}\right)^{-1}\frac{\partial}{\partial \lambda}\mathbb{E}_{q(\theta)}\left[\int dP_n(x)\left(\int p(y|x,\theta)^{1+\beta}dy\right) - \int p(y|x',\theta)^{1+\beta}dy\right].\end{aligned}\tag{4.59}$$

If we take the limit β to 0, the above expression reduced to IF of ordinary VI.

4.6.5.3 Derivation of the IF for γ VI

We can derive IF for γ VI in the same way as β VI. For simplicity, we focus on the transformed cross entropy, which is given Eq.(4.34). For unsupervised situation, the first order condition is given by

$$\begin{aligned}0 &= \frac{\partial}{\partial \lambda}L_\gamma\bigg|_{\lambda=\lambda^*}\\ &= \nabla_\lambda \mathbb{E}_{q(\theta;\lambda^*(\epsilon))}\left[N\int dP_{\epsilon,z}(x)\frac{p(x|\theta)^\gamma}{\left\{\int p(x|\theta)^{1+\gamma}dx\right\}^{\frac{\gamma}{1+\gamma}}} + \ln p(\theta) - \ln q(\theta;\lambda^*(\epsilon))\right].\end{aligned}\tag{4.60}$$

In the same way as β VI, we can get the IF of γ VI for unsupervised setting as,

$$\frac{\partial \lambda^*(\varepsilon)}{\partial \varepsilon} = -\left(\frac{\partial^2 L_\gamma}{\partial \lambda^2}\right)^{-1}\frac{\partial}{\partial \lambda}\mathbb{E}_{q(\theta)}\left[N\frac{\int dP_n(x)p(x|\theta)^\gamma - p(z|\theta)^\gamma}{\left\{\int p(x|\theta)^{1+\gamma}dx\right\}^{\frac{\gamma}{1+\gamma}}}\right].\tag{4.61}$$

For supervised situation, the first order condition is give by,

$$\begin{aligned}0 &= \frac{\partial}{\partial \lambda}L_\gamma\bigg|_{\lambda=\lambda^*}\\ &= \nabla_\lambda \mathbb{E}_{q(\theta;\lambda^*(\epsilon))}\left[N\int dP_{\epsilon,z}(x,y)\frac{p(y|x,\theta)^\gamma}{\left\{\int p(y|x,\theta)^{1+\gamma}dy\right\}^{\frac{\gamma}{1+\gamma}}} + \ln p(\theta) - \ln q(\theta;\lambda^*(\epsilon))\right].\end{aligned}\tag{4.62}$$

In the same way as β VI, we can get the IF of γ VI for supervised setting as,

$$\frac{\partial \lambda^*(\varepsilon)}{\partial \varepsilon} = -N\left(\frac{\partial^2 L_\gamma}{\partial \lambda^2}\right)^{-1}\frac{\partial}{\partial \lambda}\mathbb{E}_{q(\theta)}\left[\int dP_n(x,y)\frac{p(y|x,\theta)^\gamma}{\left\{\int p(y|x,\theta)^{1+\gamma}dy\right\}^{\frac{\gamma}{1+\gamma}}} - \frac{p(y'|x',\theta)^\gamma}{\left\{\int p(y|x',\theta)^{1+\gamma}dy\right\}^{\frac{\gamma}{1+\gamma}}}\right].\tag{4.63}$$

4.6.6 Discussion about the different definition of the IF

So far, we considered that outliers are added to the original training dataset. We can consider another type of contamination, for example, one of the observed data is perturbed. This is a situation that observed data $z = (x, y)$ is perturbed, for example, like $z_\epsilon = (x + \epsilon, y)$. We call this type of data contamination as a data perturbation. This is the contamination Koh and Liang (2017) discussed.

As for a data perturbation, the IF of ordinary VI is given as

$$\frac{\partial \lambda^* (\varepsilon)}{\partial \varepsilon} = - \left(\frac{\partial^2 L}{\partial \lambda^2} \right)^{-1} \frac{\partial}{\partial \lambda} \mathbb{E}_{q(\theta)} \left[\frac{\partial}{\partial x} \ln p \left(z | \theta \right) \right]. \tag{4.64}$$

IF of β divergence based VI is given as

$$\frac{\partial \lambda^* (\varepsilon)}{\partial \varepsilon} = - \left(\frac{\partial^2 L_\beta}{\partial \lambda^2} \right)^{-1} \frac{\partial}{\partial \lambda} \mathbb{E}_{q(\theta)} \left[\frac{\partial}{\partial x} p \left(z | \theta \right)^\beta \right]. \tag{4.65}$$

4.6.7 Discussion about another type of γ VI

So far, we used the transformed γ cross entropy, which is given in Eq.(4.33). The reason we used the transformed cross entropy instead of the original expression is that we can interpret the pseudo posterior when using the transformed cross entropy much easier than that uses original cross entropy.

In the same way as Eq.(4.16), we can derive the pseudo posterior using transformed cross entropy,

$$\begin{aligned} q(\theta) &\propto e^{N \frac{\gamma+1}{\gamma} \frac{1}{N} \sum_{i=1}^{N} \frac{p(x_i|\theta)^\gamma}{\{\int p(x|\theta)^{1+\gamma} dy\}^{\frac{\gamma}{1+\gamma}}}} p(\theta) \\ &= \left[\prod_i^N e^{l_\theta (x_i)} p(\theta) \right], \end{aligned} \tag{4.66}$$

where $l_\theta(x_i) = \frac{\gamma+1}{\gamma} \frac{p(x_i|\theta)^\gamma}{\{\int p(x|\theta)^{1+\gamma} dy\}^{\frac{\gamma}{1+\gamma}}}$. In this formulation, it is easy to consider that the information of data x_i is utilized to update the prior information through $e^{l_\theta(x_i)}$.

However, when using original cross entropy, such interpretation cannot be done because the pseudo posterior is given by,

$$q(\theta) \propto e^{N(\frac{1}{\gamma} \ln \frac{1}{N} \sum_i^N p(x_i|\theta)^\gamma dx - \frac{1}{1+\gamma} \ln \int p(x|\theta)^{1+\gamma} dx)} p(\theta), \tag{4.67}$$

Since this pseudo posterior has not additivity, it is difficult to understand how each training data x_i contributes to update the parameter. Moreover it is not straight forward to apply stochastic variational inference framework. Accordingly, we decided to use the transformed cross entropy.

Even thought the interpretation is difficult we can dirive IF in the same way as we discussed. For unsupervised situation, the first order condition is given by

$$\begin{aligned} 0 &= \frac{\partial}{\partial \lambda} L_\gamma \bigg|_{\lambda=\lambda^*} \\ &= \nabla_\lambda \mathbb{E}_{q(\theta;\lambda^*(\epsilon))} \left[\frac{N}{\gamma} \ln \int dP_{\epsilon,z}(x) p(x|\theta)^\gamma dx - \frac{N}{1+\gamma} \ln \int p(x|\theta)^{1+\gamma} dx + \ln p(\theta) - \ln q(\theta; \lambda^*(\epsilon)) \right]. \end{aligned} \tag{4.68}$$

In the same way as β VI, we can get the IF of γ VI of original cross entropy for unsupervised setting as,

$$\frac{\partial \lambda^{*}\left(\varepsilon\right)}{\partial \varepsilon}=-\frac{N}{\gamma}\left(\frac{\partial^{2} L_{\gamma}}{\partial \lambda^{2}}\right)^{-1}\frac{\partial}{\partial \lambda}\mathbb{E}_{q(\theta)}\left[\frac{\int dP_{n}(x)p(x|\theta)^{\gamma}-Np(z|\theta)^{\gamma}}{\int dP_{n}(x)p(x|\theta)^{\gamma}}\right]. \tag{4.69}$$

For a supervised situation, we can derive the estimation equation in the same way.

4.6.8 Discussion of the IF for neural networks

Here, we describe the discussion of the IF's behavior when using a neural network model for regression and classification problems with the logistic loss.

We analyze the IF of the variational parameter in the approximate posterior distribution. We use mean-field variational inference and use the Gaussian distribution for the approximate posterior distribution for the neural network parameters. $q(\theta)$ denote the approximate posterior and θ corresponds to $\{W, b\}$ in the neural network. Since $q(\theta)$ is the Gaussian distribution, we parametrize it by $\lambda = \{\mathbb{E}[\theta], \mathbb{E}[\theta\theta^{\top}]\}$, that is, the first moment and the second moment. Then the variational posterior is expressed as $q(\theta; \lambda)$. We first analyze the IF of $\mathbb{E}[\theta]$ since this is mean parameter and more important than $\mathbb{E}[\theta\theta^{\top}]$. For simplicity, λ indicates only $\lambda = \mathbb{E}[\theta]$ below. We discuss about $\mathbb{E}[\theta\theta^{\top}]$ later.

Let us start from ordinary variational inference. In Eq.(4.54), we focus on the term, $\frac{\partial}{\partial \lambda}\mathbb{E}_{q(\theta;\lambda)}\left[\ln p\left(y|\theta\right)\right]$, because this is the only term that is related to outlier. When we assume that approximate posterior is the Gaussian distribution, we can transform this term in the following way,

$$\begin{aligned}\frac{\partial}{\partial \lambda}\mathbb{E}_{q(\theta;\lambda)}\left[\ln p\left(y|\theta\right)\right] &= \frac{\partial}{\partial \lambda}\left\{\int q\left(\theta;\lambda\right)\ln p\left(y|\theta\right)d\theta\right\}\\ &= \int \frac{\partial q\left(\theta;\lambda\right)}{\partial \lambda}\ln p\left(y|\theta\right)d\theta\\ &= -\int q\left(\theta;\lambda\right)\frac{\partial}{\partial \theta}\ln p\left(y|\theta\right)d\theta\\ &= -E_{q(\theta;\lambda)}\left[\frac{\partial}{\partial \theta}\ln p\left(y|\theta\right)\right],\end{aligned} \tag{4.70}$$

where we used partial integration for the second line to third line and we also used the following relation which holds for the Gaussian distribution,

$$\frac{\partial q\left(\theta;\lambda\right)}{\partial \lambda}=\frac{\partial q\left(\theta;\lambda\right)}{\partial \theta}. \tag{4.71}$$

This relation also holds for the Student-t distribution. From the above expression, it is clear that studying the behavior of $\frac{\partial}{\partial \theta}\ln p\left(y|\theta\right)$ is crucial for analyzing the IF. In this case, the behavior of IF in this expression is similar to that of maximum likelihood.

About the parameter $\lambda = \mathbb{E}[\theta\theta^\top]$,

$$\frac{\partial}{\partial\lambda}\mathbb{E}_{q(\theta;\lambda)}\left[\ln p\left(y|\theta\right)\right] = \frac{\partial}{\partial\lambda}\left\{\int q\left(\theta;\lambda\right)\ln p\left(y|\theta\right)d\theta\right\} = \mathbb{E}_{q(\theta;\lambda)}\left[\nabla_\lambda \ln q\left(\theta;\lambda\right)\ln p\left(y|\theta\right)\right], \tag{4.72}$$

Then, outlier related term is $\ln p\left(y|\theta\right)$, thus, this term will be crucial to analyze the IF of $\lambda = \mathbb{E}[\theta\theta^\top]$.

4.6.8.1 Regression

Here, we consider the regression problem by a neural network. We denote the input to the final layer as a function $f_\theta(x)$, where x is the input data and θs are random variables which obeys approximate posterior $q(\theta;\lambda)$. We consider the output layer as the Gaussian distribution as $p(y|f_\theta(x)) = N(y|f_\theta(x),\sigma^{-1}I)$. From above discussion, what we have to consider is $\frac{\partial}{\partial\theta}\ln p\left(y|f_\theta(x)\right)$ for the analysis of $[\theta]$. We denote input related outlier as x_o, which means x_o does not follow the same distribution as the ordinary training dataset. Also, we denote the output related outlier as y_o that it does not follow the same observation noise as the ordinary training dataset.

Output related outlier

Since we consider the model of which output layer is the Gaussian distribution, following relation holds for IF of ordinary VI,

$$\frac{\partial}{\partial\theta}\ln p\left(y_o|f_\theta(x_o)\right) \propto \left(y_o - f_\theta(x_o)\right)\frac{\partial f_\theta(x_o)}{\partial\theta}. \tag{4.73}$$

We can see that this term does not bounded when $y_o \to \pm\infty$. And thus IF of ordinary VI is unbounded as output related outlier become large. About the parameter $\lambda = \mathbb{E}[\theta\theta^\top]$, $\ln p\left(y|\theta\right) \propto \left(y_o - f_\theta(x_o)\right)^2$ thus this is also not bounded.

As for the β divergence, we have to treat Eq.(4.59). Fortunately, when we use the Gaussian distribution for the output layer, the second term in the bracket of Eq.(4.59) will be constant by the analytical integration, and thus its derivative will be zero. Therefore the output related term is only the first term of Eq.(4.59). Thanks to this property, the denominator of Eq.(4.63) will also be a constant. Therefore IF of β VI and γ VI behaves in the same way. Therefore, we only consider β VI for the regression. We get the following expression,

$$\begin{aligned}\frac{\partial}{\partial\theta}p\left(y_o|f_\theta(x_o)\right)^\beta &\propto e^{-\frac{\beta}{2}(y_o - f_\theta(x_o))^2}\left(y_o - f_\theta(x_o)\right)\frac{\partial f_\theta(x_o)}{\partial\theta}\\ &= \frac{\left(y_o - f_\theta(x_o)\right)}{e^{\frac{\beta}{2}(y_o - f_\theta(x_o))^2}}\frac{\partial f_\theta(x_o)}{\partial\theta}.\end{aligned} \tag{4.74}$$

From this expression, we can see that IF of β VI is bounded because Eq.(4.74) goes to 0 as $y_o \to \pm\infty$. This means that the influence of this contamination will become zero. This is the desired property for robust estimation. About the second moment parameter $\lambda = \mathbb{E}[\theta\theta^\top]$, we can show that the output related term is $p\left(y_o|f_\theta(x_o)\right)^\beta$ and this term is proportional to $e^{-\frac{\beta}{2}(y_o - f_\theta(x_o))^2}$. Thus it is always bounded. Thus, we can say that IFs of the second moment parameter is also always bounded.

Input related outlier

Next, we consider input related outlier. We consider whether Eq.(4.73) and Eq.(4.74) are bounded or not when $x_o \to \pm\infty$. As for ordinary VI, from Eq.(4.73), if $f_\theta(x) \to \pm\infty$, the IF will diverge. For β-VI, from Eq.(4.74), even if $f_\theta(x) \to \pm\infty$, the IF will not diverge. Thus, we need to study $f_\theta(x)$ will diverge or not.

To proceed the analysis, we have to specify models.

1. We start from the linear regression, $f_\theta(x_o) = W_1 x_o + b_1$, where $\theta = \{W_1, b_1\}$. In this case $\frac{\partial f_\theta(x_o)}{\partial W_1} = x_o$ and $\frac{\partial f_\theta(x_o)}{\partial b_1} = 1$. When $x_o \to \pm\infty$, $f_\theta(x_o) \to \pm\infty$.
   From these, we can easily find that Eq.(4.73) is unbouded. As for Eq.(4.74), the exponential function in the denominator of Eq.(4.74) plays a crucial role. Thanks to this exponential function,
   $$\begin{aligned}\frac{\partial}{\partial W_1} p\left(y_o | f_\theta(x_o)\right)^\beta \propto & \frac{(y_o - f_\theta(x_o))}{e^{\frac{\beta}{2}(y_o - f_\theta(x_o))^2}} x_o \\ & \xrightarrow[x_o \to \infty]{} 0. \end{aligned} \tag{4.75}$$
   From these, ordinary VI is not robust against input related outliers, however β VI is robust.

2. Next we consider the situation that there is a hidden layer, that is $f_\theta(x_o) = W_2(W_1 x_o + b_1) + b_2$, where $\theta = \{W_1, b_1, W_2, b_2\}$. Here, we do not consider activation function and the model which has activation is described later. Following relations hold,
   $$\frac{\partial}{\partial W_1} f_\theta(x_o) = W_2 x_o, \quad \frac{\partial}{\partial W_2} f_\theta(x_o) = W_1 x_o + b_1 \tag{4.76}$$
   From these relations, the behavior of IF in the case of $x_o \to \pm\infty$ is actually as same as the case where there is no hidden layers. Therefore, IF of input related outlier is bounded in β VI and that is unbounded in ordinary VI. Even if we add more layers the situation does not change in this situation where no activation exists.

3. Next, we consider the situation that the model has an activation function. We consider $relu$ and $tanh$ as activation function. In the situation that there is only one hidden layers, $f_\theta(x_o) = W_2(relu\,(W_1 x_o + b_1)) + b_2$,
   $$\frac{\partial f_\theta(x_o)}{\partial W_2} = relu\left(W_1 x_o + b_1\right), \quad \frac{\partial f_\theta(x_o)}{\partial W_1} = \begin{cases} W_2 x_o, & W_1 x_o + b_1 \geq 0 \\ 0, & W_1 x_o + b_1 < 0, \end{cases} \tag{4.77}$$
   Actually, this is almost the same situation as when there are no activation functions, because there remains possibility that IF will diverge in ordinary VI, while IF in β VI is bounded.

   When we use $tanh$ as a activation function, $f_\theta(x_o) = W_2 tanh\left(W_1 x_o + b_1\right) + b_2$,
   $$\frac{\partial f_\theta(x_o)}{\partial W_1} = \frac{W_2 x_o}{cosh^2\left(W_1 x_o + b_1\right)} \xrightarrow[x_o \to \infty]{} 0. \tag{4.78}$$

The limit of above expression can be easily understand from Figure.4.5. From this expression,

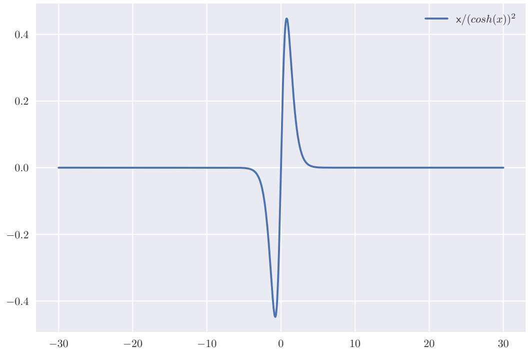


FIGURE 4.5: Behavior of $\frac{x}{\cosh^2 x}$

we can understand IF of W_1 is bounded in both ordinary estimator and β estimator, when we consider the model, $f_\theta(x_o) = tanh\,(W_1 x_o + b_1)$. As for W_2,

$$\frac{\partial f_\theta(x_o)}{\partial W_2} = tanh(W_1 x_o + b_1). \tag{4.79}$$

In this expression, even if input related outlier goes to infinity, the maximum of the above expression is 1. Accordingly, the IF of W_2 is bounded in any case. And thus IF of both ordinary VI and β VI is bounded when we use $tanh$ activation function.

4. Up to now, we have seen the model which has a hidden model. The same discussion can be held for the model which has more hidden layers. If we add layers, the above discussion holds and there remains a possibility that IF using relu in ordinary VI will diverge.

   We can say that ordinary VI is not robust to output related outliers and input related outliers. The exception is that using tanh activation function makes the IF of ordinary VI bounded. In β VI, the IF of parameters is always bounded.

About the second moment parameter $\lambda = \mathbb{E}[\theta\theta^\top]$, the similar discussion holds as the case of the output related outlier, and hence the IF in usual VI is not bounded but those in β, γ VI is always bounded.

Using the Student-t distribution for the output layer

We additionaly consider the property of the Student-t loss in stead of the Gaussian. When we denote degree of freedom as v, and the variance as σ^2, following relation holds,

$$\frac{\partial}{\partial\theta} \ln p\,(y_o|f_\theta(x_o)) \propto \frac{(y_o - f_\theta(x_o))}{v\sigma^2 + (y_o - f_\theta(x_o))^2} \frac{\partial f_\theta(x_o)}{\partial\theta}. \tag{4.80}$$

By comparing Eq.(4.80) with Eq.(4.73) and Eq.(4.74), we can confirm that the behavior of IF in the case of the Student-t loss in ordinary VI is similar to the Gaussian loss model in β VI. First, consider output related outlier,

$$\frac{\partial}{\partial \theta} \ln p\left(y_o | f_\theta(x_o)\right) \xrightarrow[y_o \to \infty]{} 0. \tag{4.81}$$

From above expression, we can find that Student-t loss is robust to output related outlier. This is the desiring property of the Student-t.

Next consider input related outlier. We consider the model, $f_\theta(x_o) = W_1 x_o + b_1$, where $\theta = \{W_1, b_1\}$

$$\begin{aligned}\frac{\partial}{\partial W_1} \ln p\left(y_o | f_\theta(x_o)\right) &\propto \frac{(y_o - f_\theta(x_o))}{v\sigma^2 + (y_o - f_\theta(x_o))^2} x_o \\ &= \frac{(y_o - f_\theta(x_o))^2}{v\sigma^2 + (y_o - f_\theta(x_o))^2} \frac{x_o}{y_o - f_\theta(x_o)} \\ &= \frac{(y_o - f_\theta(x_o))^2}{v\sigma^2 + (y_o - f_\theta(x_o))^2} \frac{f_\theta(x_o) - b_1}{W_1(y_o - f_\theta(x_o))} \\ &\xrightarrow[x_o \to \infty]{} -W_1^{-1}. \end{aligned} \tag{4.82}$$

This is an interesting result that in β VI, the IF of input related outlier goes to 0 in the limit, on the other hand for the Student-t loss, the IF is bounded but finite value remains.

4.6.8.2 Classification

Here, we consider the classification problem. We focus on the binary classification, and output y can take $+1$ or 0. We only consider the input related outlier because the influence caused by label misspecification is always bounded.

As the model, we consider the logistic regression model,

$$p(y|f_\theta(x)) = f_\theta(x)^y (1 - f_\theta(x))^{(1-y)}, \tag{4.83}$$

where

$$f_\theta(x) = \frac{1}{1 + e^{-g_\theta(x)}}, \tag{4.84}$$

where $g_\theta(x)$ is input to sigmoid function. We consider a neural net for $g_\theta(x)$.

1. We first assume $g_\theta(x) = Wx + b$, then $\frac{\partial g}{\partial W} = x$ and $\frac{\partial g}{\partial b} = 1$. Let us start from ordinary VI and consider outlier related term of it. From the relation Eq.(4.70),

$$\begin{aligned}\frac{\partial}{\partial \theta} \ln p(y|f_\theta(x)) &= \frac{\partial}{\partial \theta} \left(y \ln f_\theta(x) + (1 - y) \ln(1 - f_\theta(x))\right) \\ &= -y(1 - f) \frac{\partial g}{\partial \theta} + (1 - y) f \frac{\partial g}{\partial \theta}. \end{aligned} \tag{4.85}$$

Let us consider, for example $y=+1$

$$\frac{\partial}{\partial\theta}\ln p(y=+1|f_\theta(x))=\frac{1}{1+e^{g_\theta(x)}}\frac{\partial g}{\partial\theta}. \tag{4.86}$$

As for $\theta=b$, this is always bounded. As for $\theta=W$,

$$\frac{\partial}{\partial W}\ln p(y=+1|f_\theta(x))=\frac{1}{1+e^{Wx+b}}x. \tag{4.87}$$

In above expression, if we take limit $x\to+\infty$, and if $Wx\to-\infty$, above expression can diverge. If $Wx\to\infty$ when $x\to+\infty$, above expression goes to 0. From this observation, it is clear that there is a possibility that IF for input related outlier diverges in a logistic regression for ordinary VI.

As for β VI, we have to consider Eq.(4.59). The first term is:

$$p(y=+1|f_\theta(x))^\beta\frac{\partial}{\partial\theta}\ln p(y=+1|f_\theta(x))=\frac{1}{(1+e^{-g_\theta(x)})^\beta}\frac{1}{1+e^{g_\theta(x)}}\frac{\partial g}{\partial\theta}. \tag{4.88}$$

If we take the limit $g\to\pm\infty$, $\frac{1}{1+e^g}\frac{1}{1+e^{-g}}\to 0$. Thus, this expression converges to 0 when $x_o\to\pm\infty$.

Next, we consider the second term in Eq.(4.59), which is constant in regression. The second term of Eq.(4.59) can be written as

$$\begin{aligned}&\left(\frac{\partial^2 L_\beta}{\partial\lambda^2}\right)^{-1}\frac{\partial}{\partial\lambda}\mathbb{E}_{q(\theta)}\left[N\int p(y|x_o,\theta)^{1+\beta}dy\right]\\&=N\left(\frac{\partial^2 L_\beta}{\partial\lambda^2}\right)^{-1}\frac{\partial}{\partial\lambda}\mathbb{E}_{q(\theta)}\left[f_\theta(x_o)^{1+\beta}+(1-f_\theta(x_o))^{1+\beta}\right].\end{aligned} \tag{4.89}$$

To proceed the analysis, we can use the relation Eq.(4.70). Since the inverse of hessian matrix is not related to outlier, what we have to consider is

$$\begin{aligned}&\int d\theta q\left(\theta\right)\frac{\partial}{\partial\theta}f_\theta(x_o)^{1+\beta}+\frac{\partial}{\partial\theta}(1-f_\theta(x_o))^{1+\beta}\\&=-\int d\theta q\left(\theta\right)\left(f_\theta(x_o)^{1+\beta}(1-f_\theta(x_o))\frac{\partial g}{\partial\theta}+(1-f_\theta(x_o))^{1+\beta}f_\theta(x_o)\frac{\partial g}{\partial\theta}\right)\\&=-\int d\theta q\left(\theta\right)\left\{(1-f_\theta(x_o))^\beta+f_\theta(x_o)^\beta\right\}(1-f_\theta(x_o))f_\theta(x_o)\frac{\partial g}{\partial\theta}.\end{aligned} \tag{4.90}$$

Since in the logistic regression situation, f_θ is bounded under from 0 to 1, the term $(1-f_\theta(x_o))^\beta+f_\theta(x_o)^\beta$ is always larger than 0. Therefore, what we have to consider is the term $(1-f_\theta(x_o))f_\theta(x_o)\frac{\partial g}{\partial\theta}$. Then,

$$(1-f_\theta(x_o))f_\theta(x_o)\frac{\partial g}{\partial\theta}=\frac{1}{1+e^g}\frac{1}{1+e^{-g}}\frac{\partial g}{\partial\theta}\xrightarrow[x_o\to\infty]{}0. \tag{4.91}$$

Thus, both the first and the second term is bounded and therefore, IF of logistic regression when using β VI is bounded.

2. Consider the case where there exists activation functions such as $relu$ or $tanh$. Let us start from the ordinary VI. Since we do not use the activation function for the final layer g and $\nabla_\theta g$ are not bounded and thus, the IF of logistic regression using $relu$ activation function is not bounded. When using $tanh$ activation function, as we discussed in regression g and $\nabla_\theta g$ are bounded and thus, IF are always bounded. Accordingly, we conclude that for the logistic regression, $relu$ activation function is not robust against input related outliers when using ordinal VI, while $tanh$ activation function is robust.

   As for β VI, it is clear from Eq.(4.88) and Eq.(4.91) that IF is bounded for both relu and tanh even using neural net since $g \to \pm\infty$, $\frac{1}{1+e^g}\frac{1}{1+e^{-g}} \to 0$.

3. Next, we consider the case of γ VI. The difference from β VI is the second term of Eq.(4.63). With the relation Eq.(4.70), and the inverse of hessian matrix is not related to outlier, the outlier related term is,

$$\begin{aligned}&\int d\theta q\left(\theta\right)\frac{\partial}{\partial\theta}\frac{p(y'|x')^\gamma}{\{\int p(y|x',\theta)^{1+\gamma}dy\}^{\frac{\gamma}{1+\gamma}}}\\&=\int d\theta q\left(\theta\right)\frac{\{\int p(y|x',\theta)^{1+\gamma}dy\}^{\frac{\gamma}{1+\gamma}}\frac{\partial}{\partial\theta}p(y'|x')^\gamma - p(y'|x')^\gamma\frac{\partial}{\partial\theta}\{\int p(y|x',\theta)^{1+\gamma}dy\}^{\frac{\gamma}{1+\gamma}}}{\{\int p(y|x',\theta)^{1+\gamma}dy\}^{\frac{2\gamma}{1+\gamma}}}.\end{aligned} \tag{4.92}$$

   In the above expression, what we have to consider is the numerator. The analysis of first term can be done in the same way as Eq.(4.88). Therefore it is bounded for both relu and tanh. The second term can be analyzed in the same way as Eq.(4.90), we do not have to consider it in the limit. From above discussion, the behavior of IF for γ VI is the same as that for β VI in the limit, accordingly, it is bounded even if using $relu$ activation function. If we increase the number of layers, the same discussion holds.

4.6.9 Discussion about the comparison of β VI and γ VI

Here, we compare the proposed β VI and γ VI theoretically. Although β VI and γ VI have robustness, their robustness property is different when the proportion of contamination is large. If the proportion of contamination is large the assumption of discussion of IF does not hold because we assumed that the ϵ is near zero to derive the IF.

If the proportion of contamination is not small, other kinds of discussion are needed. Such a discussion is given in in Fujisawa and Eguchi (2008), therefore we review it and use it for our variational objectives.

Following the notation in Fujisawa and Eguchi (2008), $g(x)$ denotes the contaminated probability density function,

$$g(x) = (1-\epsilon)f(x) + \epsilon\delta(x), \tag{4.93}$$

where $f(x)$ is the underlying true probability density function, $\delta(x)$ denotes the contamination probability density function, and ϵ is the contamination proportion.

We assume that when a data point x^* is an outlier $f(x^*)$ is sufficiently small. We express this assumption by saying that the following quantity is sufficiently small for an appropriate large $\gamma_0 > 0$,

$$\nu_f = \left\{ \int \delta(x) f(x)^{\gamma_0} dx \right\}^{1/\gamma_0}. \tag{4.94}$$

This means that $\delta(x)$ exists on the tail of $f(x)$. If $\delta(x)$ is the Dirac function at x^*, $\nu_f = f(x^*)$, and above assumption simply means when a data point x^* is an outlier $f(x^*)$ is sufficiently small.

Under this assumption, following lemma and theorem holds (this is lemma3.1 and theorem 3.2 in Fujisawa and Eguchi (2008)) that

Lemma 2. *Suppose that the positive function h satisfies the above assumption, where f is replaced by h. It then holds*

$$\begin{aligned} d_\gamma(g,h) &= d_\gamma((1-\epsilon)f,h) + O(\epsilon\nu_h^\gamma) \\ &= d_\gamma(f,h) - \frac{1}{\gamma}\log(1-\epsilon) + O(\epsilon\nu^\gamma). \end{aligned} \tag{4.95}$$

Theorem 5. *Suppose that the positive function h satisfies the above assumption, where f is replaced by h. Let $\nu = \max\{\nu_f, \nu_h\}$. Then, the Pythagorean relation among g, f, and h approximately holds:*

$$\Delta(g,f,h) = D_\gamma(g,h) - D_\gamma(g,f) - D_\gamma(f,h) = O(\epsilon\nu^\gamma). \tag{4.96}$$

This theorem means that the minimizing divergence from the model h to contaminated density g is approximately equivalent to minimizing the divergence h to true distribution f and its order of error is given by $O(\epsilon\nu^\gamma)$.

Recall that the objective function of our proposed is given by

$$L_\gamma(q(\theta)) = \int q(\theta) \left(N d_\gamma \left(g(x) \| p(x|\theta)\right)\right) d\theta + D_{\mathrm{KL}}(q(\theta)\|p(\theta)), \tag{4.97}$$

where $g(x)$ is the contaminated distribution and $p(x|\theta)$ is the model we prepared. By using the Pythagorean relation, we can rewrite the above expression in the following way by using the true underlying distribution,

$$L_\gamma(q(\theta)) = \int q(\theta) \left(N d_\gamma(f(x)\|p(x|\theta)) - \frac{1}{\gamma}\log(1-\epsilon) + O(\epsilon\nu^\gamma) \right) d\theta + D_{\mathrm{KL}}(q(\theta)\|p(\theta)). \tag{4.98}$$

This equation means that by using the γ cross entropy, we can utilize the γ cross entropy between true distribution to our model. We optimized the objective function by using the black-box variational inference method and optimize the variational parameters by gradient decent, and thus the constant terms inside the integral are neglected.

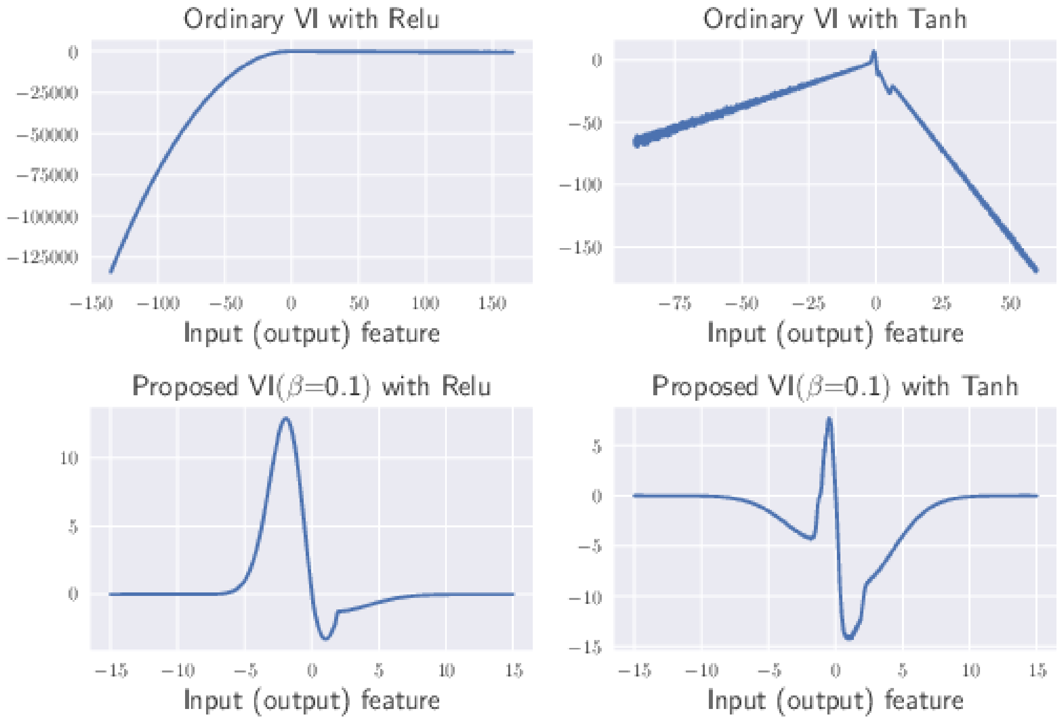


FIGURE 4.6: Perturbation on test log-likelihood for neural net regression.

This relation is obtained under the assumption of Eq. (4.94). The assumption is not the assumption that we used in the IF that contamination proportion of ϵ is small. Therefore even if the contamination proportion is large, we can obtain the The robustness of β divergence is assured by the IF (Basu et al. (1998)) and thus it is not guaranteed if the contamination proportion is not sufficiently small. Following this observation, γ divergence based method is superior to β divergence method.

4.6.9.1 Additional results of influence function experiments

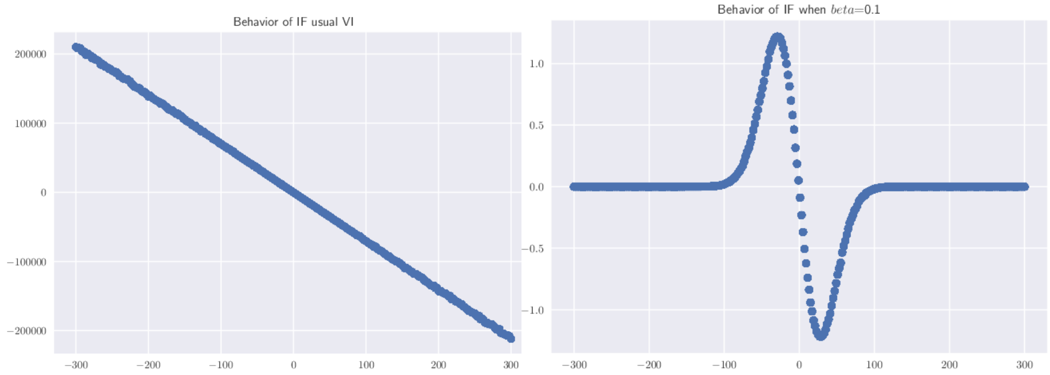
Regression

In Section 4.5.2, the figure of input and output related outlier settings are shown. Here, we show both the input and output related outlier situation. Figure. 4.6 is the case when the first feature of the input and the output value increase simultaneously.

From Figure. 4.6, we confirmed again that in this situation, the perturbation on ordinary VI is not unbounded and the perturbation on our proposed method is bounded.

IF of the parameter of the neural network

Here, we show the IF of parameters. Figure. 4.7 shows the plot of $IF(x_1, W, P)$ where W is a chosen one affine parameter in the case of $relu$ activation function (Since we consider VI, this W means the mean of the approximate posterior distribution after the optimization). Figure. 4.7(a) shows the case of ordinary VI, which diverges as absolute value of x_1 become large. This means outliers have an unlimited influence on the estimated static. On the other hand, Figure. 4.7(b) shows the case of the proposed method and the influence is bounded, that is the effect of outliers goes to zero. These results are compatible with our theoretical analysis in the previous section.



(a) VI

(b) Proposed method

FIGURE 4.7: IF of one affine parameter in Bayesian neural net.

However, this is not sufficient analysis because we would like to have robust predictive distribution. Accordingly, it is necessary to study whether the prediction is robust against outliers. For the analysis of prediction, we simulated the test log-likelihood and this is what we had seen so far. If the test log-likelihood has affected so much by an outlier, that is a prediction on the test point is affected so much. Accordingly, such a model is not robust even under the contamination of one outlier.

Chapter 5

Bayesian posterior approximation via greedy particle optimization

In this chapter, we present a new approximation approach which combines benefits of parametric approaches and sampling approaches.

5.1 Outline

In Chapters 3 and 4, we approximate the posterior distribution with parametric distributions, from which we can easily draw samples. In the parametric approximation, we often consider the mean field assumption and use an exponential family for the approximate posterior distribution (Blei et al., 2017). Since these assumptions are used to make optimization more tractable, they are often too restrictive to approximate the posterior distribution. Therefore, the approximate distribution often never converges to the posterior distribution, which means that the approximation is biased and no theoretical guarantee is assured. An alternative way is a discrete approximation of the posterior distribution by using a set of particles (Bishop, 2006), $\hat{p}(\theta) = \sum_{n=1}^{N} \delta(\theta - \theta_n)/N$. Particle approximation is free of parametric assumptions and more expressive. The Monte Carlo (MC) method is used to draw particles randomly and independently (Bishop, 2006). However, the drawback of MC is that vast computational resources are required to sample from multi-modal and high-dimensional distributions.

Recently, methods that optimize particles through iterative updates have been explored. A representative example is Stein variational gradient descent (SVGD) (Liu and Wang, 2016), which iteratively updates all particles in the direction that is characterized by kernelized Stein discrepancy (KSD). The update is actually implemented by gradient descent and SVGD empirically works well in high-dimensional problems. However, theoretical properties of SVGD have not been clarified and no finite sample bound of the convergence rate is known (Liu, 2017). Another example is the Stein points (SP) (Chen et al., 2018), which directly minimizes KSD. Although this method is assured by a finite sample convergence bound, it is not practically feasible in high-dimensional problems due to the curse of dimensionality, because gradient descent is not available and sampling or grid search needs to be used for optimization. Moreover, the number of evaluations of the gradient of the log probability, which usually requires vast computation costs, is four times that of SVGD.

We aim to develop a discrete approximation method that is computationally efficient, works well in high-dimensional problems, and also has a theoretical guarantee for the convergence rate. In this chapter, we propose *maximum mean discrepancy minimization by the Frank-Wolfe algorithm (MMD-FW)* in a greedy way. Our convex formulation of discrete approximation enables us to use the Frank-Wolfe (FW) algorithm (Jaggi, 2013) and to derive a finite sample bound of the convergence rate.

Our contributions in this chapter are three-fold:

1. We formulate a discrete approximation method in terms of convex optimization of MMD in a reproducing kernel Hilbert space (RKHS), and solve it with the FW algorithm.

2. Our algorithm is computationally efficient and empirically works well in high-dimensional problems. It has a guaranteed finite sample bound of the convergence rate.

3. We show empirically that our method compares favorably with existing particle optimization methods.

5.2 Preliminary

Here, we review two existing particle optimization methods, SVGD and SP. After that, we introduce MMD which is our objective function. We assume that $\theta \in \mathbb{R}^d$ and let $k : \Theta \times \Theta \to \mathbb{R}$ be the reproducing kernel of an RKHS $\mathcal{H}$ of functions $\Theta \to \mathbb{R}$ with the inner product $\langle \cdot, \cdot \rangle$ and $\| \cdot \|_{\mathcal{H}}$ is the assosiated norm, where $\Theta \subseteq \mathbb{R}^d$ denotes the input domain.

5.2.1 Stein variational gradient descent (SVGD)

We first prepare initial particles $\hat{p}_0(\theta) = \sum_{n=1}^{N} \delta(\theta, \theta_n)/N$ and iteratively update them by a transformation, $T(\theta) = \theta + \epsilon\phi(\theta)$, where $\phi(\theta)$ is a perturbation direction. When the current empirical distribution is $\hat{p}(\theta) = \sum_{n=1}^{N} \delta(\theta, \theta_n)/N$, then $\phi(\theta)$ is chosen to maximally decrease the Kullback-Leibler (KL) divergence between the empirical distribution $\hat{p}$ formed by the particles and the posterior distribution p,

$$\phi^*(\theta) = \underset{\phi \in \mathcal{F}}{\arg\max} \left\{ -\frac{d}{d\epsilon} \mathrm{KL}(\hat{p}_{[T]} \| p)|_{\epsilon=0} \right\}, \tag{5.1}$$

where $\mathcal{F}$ denotes a set of candidate functions from which we choose map ϕ, and

$$\hat{p}_{[T]}(\theta) = \hat{p}(T^{-1}(\theta)) \cdot |\det(\nabla_z T^{-1}(\theta))|. \tag{5.2}$$

Liu and Wang (2016) proved that this problem is characterized by the Stein operator,

$$-\frac{d}{d\epsilon} \mathrm{KL}(\hat{p}_{[\epsilon\phi]} \| p)|_{\epsilon=0} = \mathbb{E}_{\theta \sim \hat{p}}[\mathcal{S}_p \phi(\theta)], \tag{5.3}$$

Algorithm 1: Stein Variational Gradient Descent

```
Input: A posterior density $p(\theta)$ and initial particles $\{\theta_n^0\}_{n=1}^N$
Output: Particles $\{\theta_i\}_{i=1}^n$ which approximate $p(\theta)$
for iteration $l$ do
    $\theta_n^{(l+1)} \leftarrow \theta_n^{(l)} + \epsilon^{(l)} \hat{\phi}^*(\theta_n^{(l)})$, where
    $\hat{\phi}^*(\theta) = \frac{1}{N} \sum_{n=1}^N \left[ k(\theta_n^{(l)}, \theta) \nabla_{\theta_n^{(l)}} \ln p(\theta_n^{(l)}) + \nabla_{\theta_n^{(l)}} k(\theta_n^{(l)}, \theta) \right]$
end for
```

where $\mathcal{S}_p$ denotes the Stein operator

$$\mathcal{S}_p \phi(\theta) = \nabla \ln p(\theta) \phi(\theta)^\top + \nabla \cdot \phi(\theta), \tag{5.4}$$

which acts on a $d \times 1$ vector function ϕ and returns a scalar value function. Thus, the optimization problem is

$$\mathcal{S}(\hat{p} \| p) := \max_{\phi \in \mathcal{F}} \left\{ \mathbb{E}_{\theta \sim \hat{p}} [\mathcal{S}_p \phi(\theta)] \right\}. \tag{5.5}$$

The problem is how to choose an appropriate $\mathcal{F}$. Liu and Wang (2016) showed that when $\mathcal{F}$ is the unit ball in an RKHS with kernel k, the optimal map can be expressed in the following way. Let $\mathcal{H}_0$ be an RKHS defined by a kernel $k(\theta, \theta')$ and $\mathcal{H} = \mathcal{H}_0 \times \cdots \times \mathcal{H}_0$ be the $d \times 1$ vector-valued RKHS. We define $\mathcal{S}_p \otimes k(\theta, \cdot) := \nabla \ln p(\theta) k(\theta, \cdot) + \nabla_\theta k(\theta, \cdot)$, then, the optimal direction is given by

$$\phi^*_{\hat{p},p}(\cdot) = \mathbb{E}_{\theta \sim \hat{p}} [\nabla_\theta \ln p(\theta) k(\theta, \cdot) + \nabla_\theta k(\theta, \cdot)]. \tag{5.6}$$

We iteratively update particles following the above direction and obtain the empirical approximation with $\{\theta_n\}_{n=1}^N$. Theoretical analysis has been conducted in terms of the gradient flow and has shown convergence to the true posterior distribution asymptotically (Liu, 2017). However, no finite sample bound has been established. The norm of the optimal direction,

$$\mathcal{S}(\hat{p} \| p) = \| \phi^*_{\hat{p},p} \|_{\mathcal{H}} = \sqrt{\mathbb{E}_{\theta, y \sim \hat{p}} k_s(\theta, \theta')}, \tag{5.7}$$

where

$$k_s(\theta, \theta') = \nabla_\theta \nabla_{\theta'} k(\theta, \theta') + \nabla_\theta k(\theta, \theta') \nabla_{\theta'} \ln p(\theta') + \nabla_{\theta'} k(\theta, \theta') \nabla_\theta \ln p(\theta) + k(\theta, \theta') \nabla_\theta \ln p(\theta) \nabla_{\theta'} \ln p(\theta'), \tag{5.8}$$

is called kernelized stein discrepancy (KSD) (Liu et al., 2016). In summary, the algorithm of SVGD is shown in Alg.1.

5.2.2 Stein points(SP)

SP (Chen et al., 2018) minimizes the above KSD directly. When q is given by a discrete approximation $\hat{p} = \sum_{n=1}^{N} \delta(\theta, \theta_n)/N$, KSD can be written as

$$\mathcal{S}(\hat{p}\|p) = \sqrt{\sum_{i,j=1}^{N} k_s(\theta_i, \theta_j)}. \tag{5.9}$$

In SP, to obtain the n-th particle, we solve

$$\arg\min_{\theta} \sum_{i=1}^{n-1} k_s(\theta_i, \theta) \quad \text{or} \quad \arg\min_{\theta} \sum_{i=1}^{n-1} k_s(\theta_i, \theta) + k_s(\theta, \theta)/2. \tag{5.10}$$

To solve these problems, Chen et al. (2018) proposed using sampling methods or grid search. However, those methods are not applicable to high-dimensional problems due to the curse of dimensionality. Although an alternative way is to use gradient descent, this is computationally difficult in high-dimensional problems since this method needs to calculate the Hessian at each iteration. Moreover, the computation cost for evaluating the derivative of the log probability is 4 times compared to SVGD. An advantage of this method is that a finite sample convergence bound is assured theoretically.

5.2.3 Maximum mean discrepancy (MMD)

SVGD and SP use KSD as the direction of the update and the objective function. In our proposed method, we use MMD as the objective function. MMD is a kind of the worst-case error between expectations. For a given test function f, we express the integral with respect to the true posterior distribution p as $Z_{f,p} = \int f(\theta)p(\theta)d\theta$. We denote an approximation of $Z_{f,p}$ as $Z_{f,\hat{p}}$, where p is approximated by $\hat{p}$ in the same way as Eq. (1). From here, we consider the weighted empirical distribution $\hat{p}(\theta) = \sum_{n=1}^{N} w_n \delta(\theta, \theta_n)$, where w_n are the weights of each particle. Then MMD (Gretton et al., 2012) is defined as

$$\begin{aligned} \mathrm{MMD}(\{w_i, \theta_i\}_{i=1}^{N})^2 &:= \frac{1}{2} \sup_{f \in \mathcal{H}: \|f\|_{\mathcal{H}}=1} \left| Z_{f,p} - \sum_{i=1}^{N} w_i f(\theta_i) \right|^2 \\ &= \frac{1}{2} \|\mu_p - \mu_{\hat{p}}\|_{\mathcal{H}}^2 \\ &= \frac{1}{2} \left\| \mu_p - \sum_{i=1}^{N} w_i k(\theta_i, \cdot) \right\|_{\mathcal{H}}^2, \end{aligned} \tag{5.11}$$

where $\mu_p = \int k(\cdot, \theta)p(\theta)d\theta \in \mathcal{H}$ and we introduce the coefficient $\frac{1}{2}$ for convenience in later calculation. We also express $\mathrm{MMD}(\{w_i, \theta_i\}_{i=1}^{N})^2$ as $\mathrm{MMD}(\mu_{\hat{p}})^2$ for simplicity.

```
Algorithm 2: Frank-Wolfe (FW) Algorithm
1: Let $\theta_0 \in \mathcal{D}$
2: for $n = 0, \dots, N$ do
3:     Compute $s = \mathrm{argmin}_{s \in \mathcal{D}} \langle s, \nabla f(\theta_n) \rangle$
4:     Constant step: $\lambda_n = \frac{1}{n+1}$
5:     Update $\theta_{n+1} = (1 - \lambda_n)\theta_n + \lambda_n s$
6: end for
```

5.3 Proposed methods

Here, we formally develop our MMD-FW. We will introduce the FW algorithm in an RKHS, propose our MMD-FW, and give a finite sample convergence bound of our method.

5.3.1 MMD minimization by the FW algorithm (MMD-FW)

On the basis of the existing methods reviewed in Section 5.2, we would like to obtain a method to approximate the posterior by discrete particles, which has high computational efficiency and theoretical guarantee. The key idea is to perform discrete approximation by minimizing MMD, instead of KSD since it causes computational problems as we described above. We minimize $\mathrm{MMD}(\mu_{\hat{p}})^2 = \frac{1}{2}\|\mu_p - \mu_{\hat{p}}\|^2_{\mathcal{H}}$, introduced by Eq. (5.11), in a greedy way. Since this is a convex function in an RKHS, we can use the FW algorithm.

The FW algorithm, also known as the conditional gradient method (Jaggi, 2013), is a convex optimization method. It focuses on the problem $\min_{\theta \in \mathcal{D}} f(\theta)$, where f is a convex and continuous differentiable function and $\mathcal{D}$ is the domain of the problem, which is also convex. As the procedure is shown in Alg. 2, the FW algorithm optimizes the objective in a greedy way. In each step, we solve the linearization of the original f at the current state θ_n as shown in Line 3 of Alg. 2. This step is often called the linear minimization oracle (LMO). The new state θ_{n+1} is obtained by a convex combination of the previous state θ_n and the solution of the LMO, s, in Line 6 of Alg. 2. The common choice of the coefficient of the convex combination is the constant step or the line search.

Bach et al. (2012) and Briol et al. (2015) clarified the equivalence between kernel herding (Chen et al., 2010) and the FW algorithm for MMD. In our situation, we minimize MMD on the marginal polytope $\mathcal{M}$ of the RKHS $\mathcal{H}$, which is defined as the closure of the convex hull of $k(\cdot, \theta)$. We also assume that all sample points θ_i are uniformly bounded in the RKHS, i.e., for any sample point θ_i, $\exists r > 0 : \|k(\cdot, \theta)\|_{\mathcal{H}} \leq r$.

By applying the FW algorithm, we want to obtain $\mu_{\hat{p}}$ which minimizes the objective $\mathrm{MMD}(\mu_{\hat{p}})^2 = \frac{1}{2}\|\mu_p - \mu_{\hat{p}}\|^2_{\mathcal{H}}$. We express the solution after n-steps FW algorithm as $\mu^n_{\hat{p}} = \sum_{i=1}^n w^n_i k(\cdot, \theta_i)$, where $\{\theta_i\}_{i=1}^n$ are the particles and w^n_i denote the weights of the i-th particle at the n-th iteration. We can obtain $\{\theta_i\}_{i=1}^n$ in a greedy way by the FW algorithm. The method of deriving the weights are discussed later.

The LMO calculation in each step is

$$\mathrm{argmin}_{g \in \mathcal{M}} \langle \mu^n_{\hat{p}} - \mu_p, g \rangle. \tag{5.12}$$

It is known that the minimizer of a linear function in a convex set is one of the extreme points of the domain (Bach et al., 2012), and thus we derive

$$\begin{aligned}\arg\min_{g\in\mathcal{M}}\langle\mu_{\hat{p}}^{n}-\mu_p,g\rangle &= \arg\min_{\theta}\langle\mu_{\hat{p}}^{n}-\mu_p,k(\cdot,\theta)\rangle\\ &= \arg\min_{\theta}\sum_{i=1}^{n}w_i^n k(\theta_i,\theta)-\mu_p(\theta). \qquad (5.13)\end{aligned}$$

We solve this LMO by gradient descent. We initialize each θ to prepare $g=k(\cdot,\theta)$ in LMO by sampling it from the prior distribution. Since the objective of LMO is non-convex, we cannot obtain the global optimum by gradient descent in general. Fortunately, even if we solve LMO approximately, FW enables us to establish a finite sample convergence bound (Locatello et al., 2017a; Jaggi, 2013; Lacoste-Julien et al., 2013; Lacoste-Julien and Jaggi, 2015; Locatello et al., 2017b). In such an approximate LMO, we set the accuracy parameter $\delta\in(0,1]$ and consider the following approximate problem which returns approximate minimizer $\tilde{g}$ of Eq.(5.13) instead of the original strict LMO:

$$\begin{aligned}\langle\mu_{\hat{p}}^{(n)}-\mu_p,\tilde{g}\rangle &= \delta\min_{g\in\mathcal{M}}\langle\mu_{\hat{p}}^{(n)}-\mu_p,g\rangle\\ &= \delta\min_{\theta}\sum_{i=1}^{n}w_i^n k(\theta_i,\theta)-\mu_p(\theta). \qquad (5.14)\end{aligned}$$

This kind of relaxation of the LMO has been widely used and shown to be reliable (Locatello et al., 2017a; Jaggi, 2013; Lacoste-Julien et al., 2013; Lacoste-Julien and Jaggi, 2015; Locatello et al., 2017b), which is much easier to solve than the original strict LMO. We call this step Approx-LMO, and we will use gradient descent to solve Approx-LMO. The derivative with respect to θ when we use the symmetric kernel k can be written as follows:

$$\nabla_\theta\langle\mu_{\hat{p}}^{(n)}-\mu_p,g\rangle \approx \frac{1}{n}\sum_{i=1}^{n}w_i^{(n)}\left(\nabla_\theta k(\theta_i,\theta)+k(\theta,\theta_i)\nabla_{\theta_i}\ln p(\theta_i)\right). \qquad (5.15)$$

The derivation of Eq.(5.15) is given in Section 5.6.1. Using this gradient, we solve Eq.(5.14). As repeatedly pointed out in Locatello et al. (2017a); Jaggi (2013); Lacoste-Julien et al. (2013); Lacoste-Julien and Jaggi (2015); Locatello et al. (2017b), an approximate solution of the LMO is enough to assure the convergence which we describe later. For this reason, we will use gradient descent in our algorithm and also a rough estimate of the gradient is enough in our situation. A similar technique has also been discussed in Locatello et al. (2017a).

For the FW algorithm, we have to specify the initial particle θ_1 and the step size choice of the algorithm. We found that the initial particle θ_1 by the MAP estimation or approximate MAP estimation shows good performance empirically and it is recommended to prepare θ_1 as a near MAP point (we will discuss other choices in Section 5.6.2). In this approach, the constant step size and line search are not recommended because those methods uniformly reduce the weights of all the particles which has already been obtained. When we use θ_1 as a near MAP point, it is located near the highest probability mass regions, and thus we should not reduce its weight uniformly. Based on this observation, we set the step size in the same way as the fully corrective Frank-Wolfe

Algorithm 3: Approx-LMO

```
Input: $\mu_{\hat{p}}^{(n)}$
Output: $k(\cdot, \theta^{L+1})$
Prepare $g^0 = k(\cdot, \theta^0)$ where $\theta$ is initialized by
randomly or sample from prior
for $l = 0 \dots L$ do
    Compute $\nabla_\theta \langle \mu_{\hat{p}}^{(n)} - \mu_p, g^l \rangle$ by Eq.(5.15)
    Update $\theta^{(l+1)} \leftarrow \theta^{(l)} + \epsilon^{(l)} \cdot \nabla_\theta \langle \mu_{\hat{p}}^{(n)} - \mu_p, g^i \rangle$
end for
```

Algorithm 4: MMD minimization by Frank-Wolfe algorithm (MMD-FW)

```
Input: A posterior density $p(\theta)$
Output: A set of particles $(\{w_i, \theta_i\}_{i=1}^N)$
Calculate approximate MAP estimation for $\mu_{\hat{p}}^{(1)}$
for $n = 2 \dots N$ do
    $k(\cdot, \theta_n)$ =Approx-LMO($\mu_{\hat{p}}^{(n-1)}$)
    Empirical BQ weight: $\hat{w}_i^n = \sum_{m=1}^n \hat{z}_m K_{im}^{-1}$, $\hat{z}_m = \sum_{l=1}^n k(\theta_l, \theta_m)/n$
    Update $\mu_{\hat{p}}^{(n+1)} = \sum_{i=1}^n \hat{w}_i^n k(\theta, \theta_i)$
end for
```

algorithm (Lacoste-Julien and Jaggi, 2015), this method calculates all the weights at each iteration, and we can circumvent the above problem. For full correction, we use the Bayesian quadrature (BQ) weight (Huszár and Duvenaud, 2012), $w_i = \sum_m z_m K_{im}^{-1}$, where K is the Gram matrix, $z_m = \int k(\theta, \theta_m) p(\theta) d\theta$, and we approximately compute the integral with particles. Since we use the empirical approximation, this makes the convergence rate slower. We will analyze the effect of this inexact step size later.

To summarize, our proposed algorithms are given in Alg. 3 and Alg. 4, which greedily increase the number of particles whithin the FW framework to minimize MMD.

5.3.2 Theoretical guarantee

First, we describe the condition of the approximated BQ weights for the convergence rate. This is necessary condition for the theoretical guarantee of the particle approximation when the finite dimensional kernel is used in our algorithm.

Theorem 6. (Approximate step size) *In Alg. 4 at the n-th iteration, let β_i^n be the ratio between $\hat{z}_i^n$ and z_i^n, i.e., $\beta_i^n = \hat{z}_i^n / z_i^n$. When $\mathcal{H}$ is finite dimensional, if*

$$\int k(\theta, \theta) p(\theta) p(\theta') d\theta d\theta' - \sum_{i,j=1}^{n} \beta_i^n \beta_j^n z_i^n K_{ij}^{-1} z_j^n > 0 \tag{5.16}$$

holds, then Theorems 7 and 8 hold. When $\mathcal{H}$ is infinite dimensional, no condition about the approximation of the weight is needed for Theorems 7 and 8 to hold.

In Eq.(5.16), since $\int k(\theta, \theta) p(\theta) p(\theta') d\theta d\theta'$ is determined by the choice of the kernel and $p(\theta)$ and $\int k(\theta, \theta) p(\theta) p(\theta') d\theta d\theta' - \sum_{i,j=1}^n z_i^n K_{ij}^{-1} z_j^n > 0$ holds, thus β_i^n should be in some *moderate* range to satisfy the condition of Eq.(5.16). More intuitively, this condition states that if the deviation of the empirical estimate of BQ weights from the true ones is below a certain criterion, then convergence guarantee of the algorithm still holds even if the step size is inexact. The range of the moderate deviation is determined by the kernel and $p(\theta)$. The proof is given in Section 5.6.5. We also analyzed the effect of inexact step size in line search and the result states that if the ratio is between 0 to 2, then the linear convergence holds; see Section 5.6.4 for details.

Next, we state the theoretical guarantee of our algorithm. We obtain $\hat{p}(\theta) = \sum_{n=1}^{N} w_n \delta(\theta, \theta_n)$ by Alg. 4 which approximates the true posterior $p(\theta)$. Let f be the test function, then we can bound the error $|Z_{f,p} - Z_{f,\hat{p}}| = |\int f(\theta)p(\theta)d\theta - \sum_{i=1}^{N} w_i f(\theta_i)|$ as follows:

Theorem 7. (Consistency) *Under the condition of Theorem 6, the error $|Z_{f,p} - Z_{f,\hat{p}}|$ of Alg. 4 is bounded at the following rate:*

$$|Z_{f,p} - Z_{f,\hat{p}}| \leq \mathrm{MMD}(\{(w_n, \theta_n)\}_{n=1}^{N}) \leq \begin{cases} \sqrt{2} r e^{-\delta_{BQ} \frac{R^2 \delta^2 N}{2r^2}} & \textit{if } \mathcal{H} \textit{ is finite dimensional,} \\ \sqrt{\frac{(\delta_{BQ}\delta+1)2^2 r^2}{\delta(N\delta_{BQ}\delta+2)}} & \textit{if } \mathcal{H} \textit{ is infinite dimensional,} \end{cases} \tag{5.17}$$

where r is the diameter of the marginal polytope $\mathcal{M}$, δ is the accuracy parameter of the LMO, and R is the radius of the smallest ball centered at μ_p included $\mathcal{M}$ (R is strictly above 0 only when the dimension of $\mathcal{H}$ is finite). δ_{BQ} denote the error caused by the empirical approximation of the BQ weights; for details, please see Section 5.6.3.

A proof of Theorem 7 can be found in Section 5.6.3. Moreover, on the basis of the Bayesian quadrature, we can regard $Z_{f,\hat{p}}$ as the posterior distribution of the Gaussian process (Huszár and Duvenaud, 2012) (see Section 5.6.13 for details) and assure the posterior contraction rate (Briol et al., 2015). Intuitively, the posterior contraction rate indicates how fast the probability of the estimated parameter residing outside a specified region (which includes the true parameter) decreases when the size of the region is increased.

Theorem 8. (Contraction) *Let $S \subseteq \mathbb{R}$ be an open neighborhood of the true integral $Z_{f,p}$ and let $\gamma = \inf_{r' \in S^c} |r' - Z_{f,p}| > 0$. Then the posterior probability on $S^c = \mathbb{R} \setminus S$ vanishes at the following rate:*

$$\mathrm{prob}(S^c) \leq \begin{cases} \frac{2r}{\sqrt{\pi}\gamma} e^{-\delta_{BQ} \frac{R^2 \delta^2 N}{2r^2} - \frac{\gamma^2}{4r^2} e^{\delta_{BQ} \frac{R^2 \delta^2 N}{r^2}}} & \textit{if } \mathcal{H} \textit{ is finite dimensional,} \\ \sqrt{\frac{2}{\pi}} \sqrt{\frac{(\delta_{BQ}\delta+1)2^2 r^2}{\delta(N\delta_{BQ}\delta+2)}} e^{-\frac{\gamma^2}{2} \frac{\delta(N\delta_{BQ}\delta+2)}{(\delta_{BQ}\delta+1)2^2 r^2}} & \textit{if } \mathcal{H} \textit{ is infinite dimensional,} \end{cases} \tag{5.18}$$

where r is the diameter of the marginal polytope $\mathcal{M}$, δ is the accuracy parameter, and R is the radius of the smallest ball centered at μ_p that includes $\mathcal{M}$. δ_{BQ} denotes the error caused by the empirical approximation of the BQ weights; for details, please see Section 5.6.6.

In the proposed method, kernel selection is crucial both numerically and theoretically. In the above convergence proof, linear convergence occurs only under the assumption that there exists a ball with centered at μ_p whose radius R is positive within the affine hull $\mathcal{M}$. Bach et al. (2012) proved that, for infinite dimensional RKHSs, such as the case of radial basis function (RBF) kernels, such an assumption never holds. Thus, we can only have sub-linear convergence for RBF kernels in general. However, as pointed out by Briol et al. (2015) , even if we use RBF kernels, thanks to finite-precision rounding error in computers, we are treating in simulations are actually essentially finite dimensional. This also holds in our situation, and in experiments, we empirically observed the linear convergence of our algorithm. We will show such a numerical result later.

A theory for the constant step size and line search step size are shown in Section 5.6.2.

5.3.3 Discussion

For specifying the initial particle θ_1, we can sample it from the prior distribution. The merit of this approach is that we can choose the step size in a computationally less demanding way such as the constant step size and line search (shown in Section 5.6.2) since the initial particle is not in a high probability mass region, uniformly decreasing less important weights by constant step size or line search. However, we empirically found in our preliminary experiments that this initialization does not perform well compared to MAP initialization. We suspect that the gradient of Eq.(5.15) is too inexact when initial particles are sampled from the prior.

Let us analyze the reason why MAP initialization performs well as follows. Although the gradient is incorrect, the LMO can be solved with error to some extent because the first particle is close to the MAP estimation and the evaluation points of the expectation include, at least, a high density region on $p(\theta)$. If the LMO is δ-close to the true value, the weights of old incorrect particles will be updated to be small enough to be ignored as the algorithm proceeds. For such a reason, the framework using processed particles works.

The empirical approximation of the BQ weights can also be justified almost in the same way as above. Since the empirical distribution includes, at least, a high density region on $p(\theta)$, the deviation of the step size (e.g., error due to the empirical approximation) from the exact BQ weight is smaller than the criterion in Theorem 6.

In summary, since we prepare the initial particles at a high probability mass region, the FW algorithm successfully finds the next particle even though the gradient for LMO or weights are inexact. As the algorithm proceeds, the weights of less reliable particles become small and accuracy of the estimation is increased. This is an intuition how the proposed algorithm works.

5.4 Related works

Here, we discuss the relationship between our method and SVGD, SP and variational boosting.

5.4.1 Relation to SVGD

SVGD is a method of optimizing a fixed number of particles simultaneously. On the other hand, MMD-FW is a greedy method adding new particles one per step. Both methods can work in high-dimensional problems since they use the information of the gradient of the score function.

To approximate a high-dimensional posterior distribution, we may need many particles, but it is unclear how many particles are needed beforehand. Thus, a greedy approach is preferable for high-dimensional problems. Since in SVGD it is unclear how we can increase the number of particles after we finish the optimization, MMD-FW is more convenient in such a case. However, simultaneous optimization is sometimes computationally more efficient and show better performance compared to a greedy approach(see the experimental results).

Based on this fact, we combine SVGD and MMD-FW by focusing on the fact that the update equations of SVGD and MMD-FW are almost the same except for the weights. More specifically, we prepare particles by SVGD first, and then apply MMD-FW by treating particles obtained by SVGD as the initial state of each greedy particle. This combination enables us to enjoy the efficient simultaneous optimization of SVGD and the greedy property and theoretical guarantee of MMD-FW. The detailed explanation is in Section 5.6.11.

In terms of computation costs, SVGD is $\mathcal{O}(N^2)$ per iteration. In MMD-FW, we only optimize one particle, and thus, its computation cost is $\mathcal{O}(N)$ at each step inside Approx-LMO . Up to the N-th particle, the total cost is $\mathcal{O}(N(N+1)/2)$, which is in the same order as SVGD. However, the number of LMO iterations in MMD-FW is much smaller than that of SVGD since the problem involves only one particle in MMD-FW, which is much easier to solve than SVGD which treats N particles simultaneously. Therefore, we can expect the computation cost of MMD-FW to be cheaper than SVGD.

5.4.2 Relation to SP

The biggest difference between MMD-FW and SP is the objective function. Due to this difference, we use gradient descent to obtain new particles which is still computationally effective in high-dimensional problems. However, SP minimizes KSD, so we cannot use gradient descent since the calculation of the gradient requires evaluations of the Hessian at each step, which is impossible in high-dimensional problems. To cope with this problem, SP uses sampling or grid search for optimization, which does not work in high-dimensional problems due to the curse of dimensionality. As we will see later, SP does not work well with complex models (see the experimental results of SP).

Another difference is that our method can reliably use an approximate step size for the weights of particles. We have shown how the deviation of the approximate weights from the exact ones affects the convergence rate, which justified the use of our method even when the exact step size is unavailable.

Lastly, we use FW to establish a greedy algorithm. This enables us to utilize many useful variants of the FW algorithm. For details, see Section 5.6.12.

However, compared with SP, we cannot evaluate the objective function directly, so we resort to other performance measures such as the log likelihood, accuracy, or RMSE in test datasets. For SP, we can directly evaluate KSD at each iteration.

5.4.3 Relation to variational boosting

The proposed method is closely related to variational boosting (Locatello et al., 2017a). In Locatello et al. (2017a), the authors analyzed the variational boosting by using the FW algorithm and showed the convergence to the posterior distribution. In variational boosting, a mixture of Gaussian distributions are used as an approximate posterior and its flexibility is increased the number of components in the mixture of Gaussian distributions. An intuition behind the convergence of variational boosting is that any distribution can be expressed by appropriately combining Gaussian mixture distributions. That situation is quite similar to MMD-FW, where we increase the number of particles greedily. In MMD-FW, we can regard each particle as being corresponding to each component of variational boosting. In both methods, the flexibility of the approximate posterior grows as we increase the number of components or particles and this allows us to establish the linear convergence under certain conditions. The difference is that we consider the solution in an RKHS and minimize MMD to approximate the posterior for MMD-FW, while variational boosting minimizes the KL divergence and treats the posterior in the parameter space.

5.4.4 Relation to kernel herding and Bayesian quadrature

In this chapter, we assume that $p(\theta)$ is the posterior distribution. On the other hand, if $p(\theta)$ is a prior distribution, kernel herding (Chen et al., 2010) or Bayesian quadrature (Ghahramani and Rasmussen, 2003), are useful. In those methods, θ_n's are decided to directly minimize some criterions. For example, the kernel herding method (Chen et al., 2010; Bach et al., 2012) minimizes MMD in a greedy way. The biggest difference from our method is that if $p(\theta)$ is the prior distribution, we can sample many particles from $p(\theta)$ and thus we can only choose the best particle that decreases the objective function maximally at each iteration. In MMD-FW, on the other hand, we cannot prepare the particles beforehand, and thus, we directly derive particles by gradient descent.

Other related work

Recently, there has been a tendency to combine an approximation of the posterior with optimization methods, which assures us of some theoretical guarantee, e.g, Locatello et al. (2017a); Dai et al. (2016). Our approach also performs discrete approximation by convex optimization in an RKHS. Another related example is sequential kernel herding (Lacoste-Julien et al., 2015). They applied the FW algorithm to particle filtering in state space models. While their method focused on the state space models, our proposed method is a general approximation method for Bayesian inference.

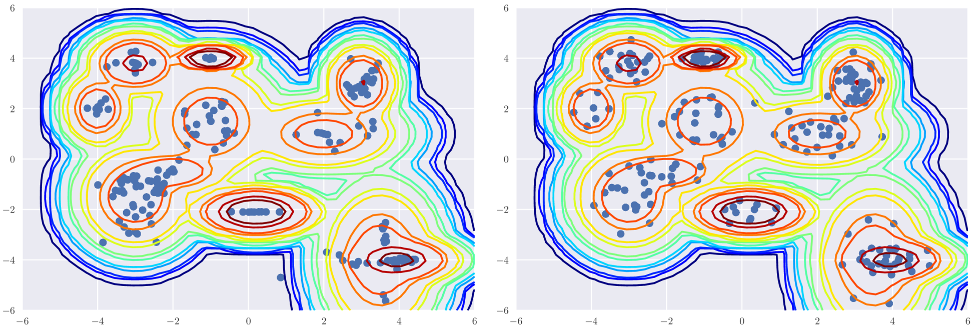
5.5 Numerical experiments

We experimentally confirmed the usefulness of the proposed method compared with SVGD and SP in both toy datasets and real world datasets. Other than comparing the performance measured in terms of the accuracy or RMSE of the proposed method with SVGD and SP, we also have the following two purposes for the experiments. The first purpose of the experiments is to confirm that our algorithm is faster than SVGD in terms of wall clock time. This is because, as mentioned before in the section of relation to SVGD, it solves simple problems compared with SVGD, thus we need less number of iterations to optimize each particle than that of SVGD in Section 5.4.1. The second purpose is to confirm the convergence behavior.

In all experiments, we used the radial basis function kernel, $k(\theta,\theta') = \exp(-\frac{1}{2h^2}\|\theta-\theta'\|_2^2)$ for proposed method and SVGD, where h is the kernel bandwidth. The choice of h is critical to the success of the algorithms. There are three methods to specify the bandwidth, fixed bandwidth, median trick, and the gradient descent. We experimented on the above three choices and found that a fixed kernel bandwidth and the median trick are stable in general, and thus, we only show the results obtained by the median trick here. For the kernel of SP, we used the three kernels proposed by the original paper (Chen et al., 2018): IMQ kernel $k_1(\theta,\theta') = (\alpha + ||\theta-\theta'||_2^2)^\beta$, inverse log kernel $k_2(\theta,\theta') = (\alpha + \log(1+||\theta-\theta'||_2^2))^{-1}$, and IMQ score kernel $k_3(\theta,\theta') = (\alpha + ||\nabla \log p(\theta) - \nabla \log p(\theta')||_2^2)^\beta$, where $\alpha = 1.0$ and $\beta = 0.5$ are used as suggested in the original paper. For the approx-LMO, we used Adam (Kingma and Ba, 2014) for all experiments.

Toy data

To clarify how our method works, we applied our algorithm to a two dimensional toy dataset and observed how the particles approximate the target distribution. The true distribution is a two dimensional mixture of Gaussians which is composed of 11 Gaussian distributions. First, we studied the results of MMD-FW and SVGD by median trick visually and the result is shown in Figure. 5.1. In the figure, the target distribution is represented in contour and red lines mean the high probability mass regions and on the other hand, blue lines mean the low probability mass regions.



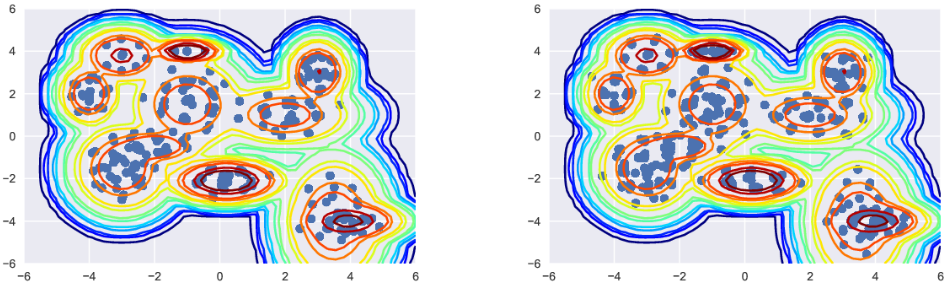
(a) 2D gaussian with particles obtained by MMD-FW

(b) 2D gaussian with particles obtained by SVGD

FIGURE 5.1: Toy data example results of MMD-FW and SVGD by the median trick

We also visualized how the choice of the bandwidth affects the results. In Fig 5.3, we used the fixed bandwidth in MMD-FW. As shown in Fig 5.3(a), the small bandwidth $h = 0.1$ makes the particles scattered. This is because the second term of the update equation, which corresponds to the entropy term, becomes very large due to the small bandwidth. When we use a large bandwidth $h = 1.0$, the results is shown in Fig 5.3(b) and particles are collapsed to modes. This is because the entropy term becomes small, and thus the repulsion force between particles become small (see Section 5.6.1 for the detailed explanation).

We also observed how the positions of particles change as we increase the total number of particles. In Figure. 5.2, we visualized the results when 300 and 500 particles were used. From the figure, as we increase the number of particles, the particles are more dispersed.

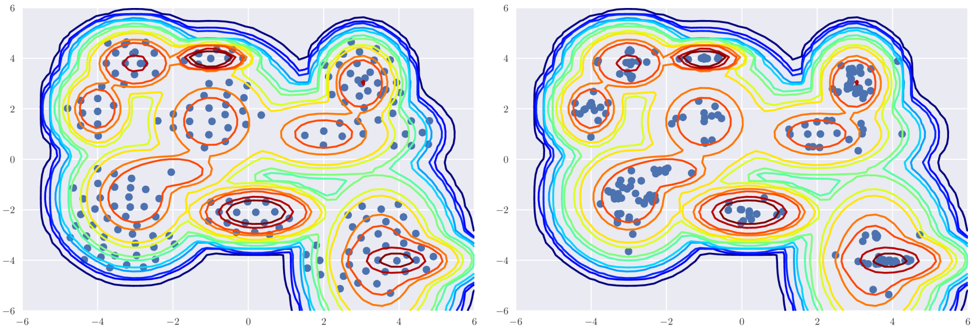


(a) 2D gaussian with 300 particles obtained by MMD-FW

(b) 2D gaussian with 500 particles obtained by MMD-FW

FIGURE 5.2: Toy data example results of MMD-FW by the median trick

Finally, by changing the number of particles and L, which is the number of gradient descent in the approx-LMO, we studied how the final MMD changes. The result is shown in Fig 5.4. For the comparison, the result of SVGD is also shown in the figure. We found that both in MMD-FW and SVGD, MMD decreases as we increase the number of particles, while the number of L does not affect MMD so much compared to the number of the particles.



(a) 2D gaussian with particles obtained by MMD-FW with fixed bandwidth $h = 0.1$

(b) 2D gaussian with particles obtained by MMD-FW with fixed bandwidth $h = 1.0$

FIGURE 5.3: The results of the toy data by MMD-FW of fixed bandwidth

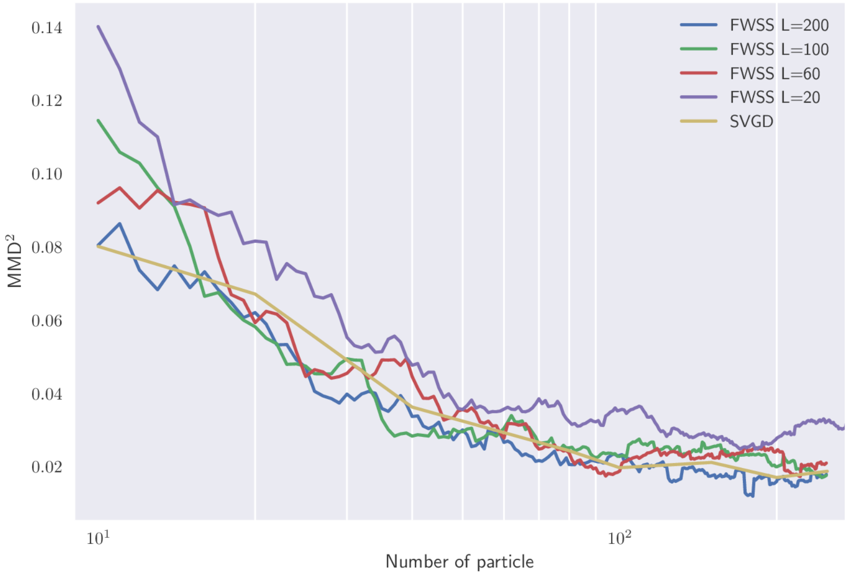


FIGURE 5.4: Final MMDs changing the number of particles and L

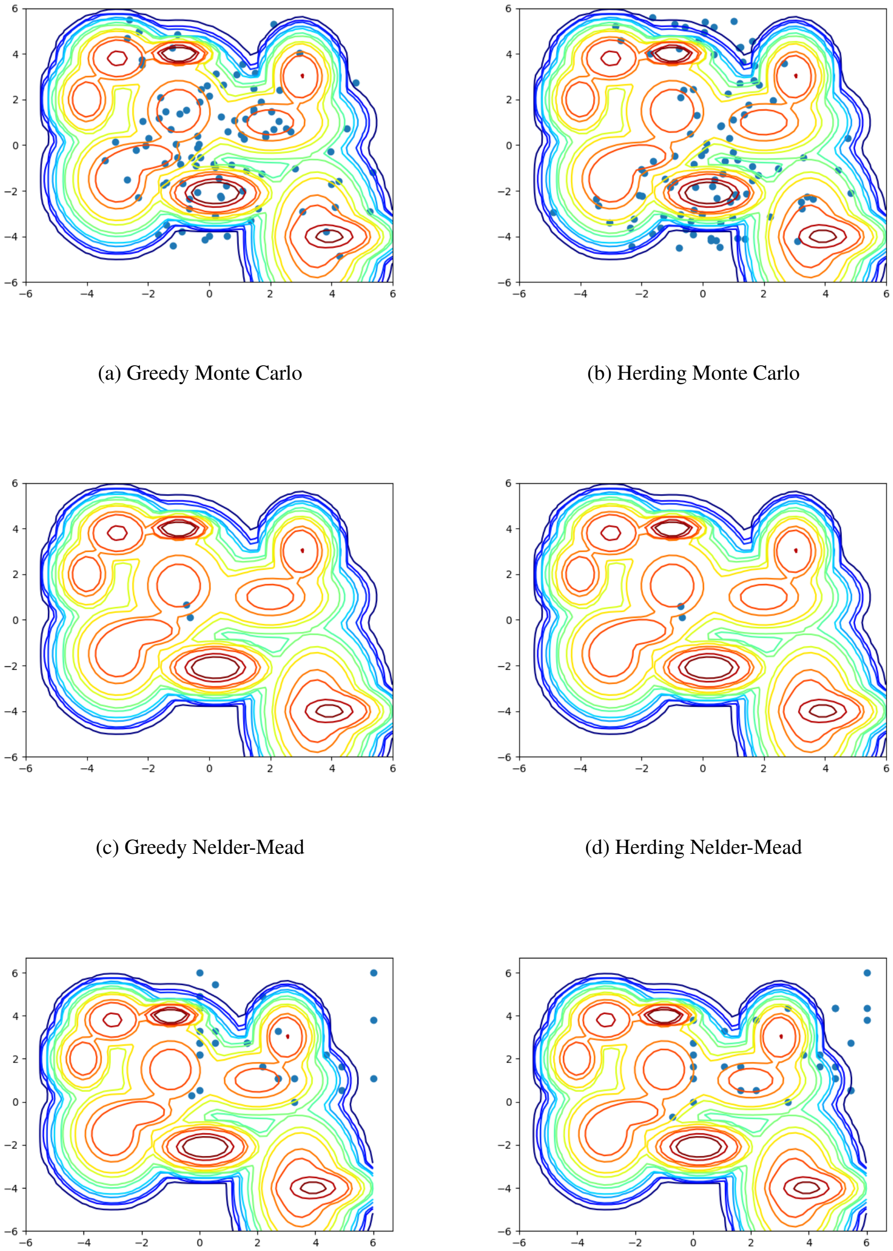
Stein Points Experiments

Next, we applied the stein points (SPs) to the toy dataset. SPs utilizes two algorithms for the objective function, the greedy algorithm and the herding algorithm. Chen et al. (2018) proposed 3 methods for the optimization: the Nelder-Mead method, the Monte Carlo method and the grid search method. Thus, we conducted experiments in 6 different combinations. Detailed explanation of the experimental settings of SP is shown in Section 5.6.8. The result shown in Figure. 5.5 is not favorable as expected. As the figure shows SPs failed to capture the character of the posterior distribution since it only does exploration. Since the greedy algorithm together with Monte Carlo seems to perform the best fit, we use this setting in the Bayesian logistic regression experiment in the next section.

We also tried to test the SP method on Bayesian neural network settings. However, it is not realistic since the dimension of the parameter space is too large.

Bayesian logistic regression

We considered Bayesian logistic regression for binary classification. The settings were the same as in those Liu and Wang (2016), where we put a Gaussian prior $p_0(w|\alpha) = N(0, \alpha^{-1})$ for regression weights w and $p_0(\alpha) = \mathrm{Gamma}(1, 0.01)$. As the dataset, we used Covertype (Dheeru and Karra Taniskidou, 2017), with 581,012 data points and 54 features. The posterior dimension is 56. In this experiment, we used Adam with a learning rate of 0.005 and we split the data, 90% are used for training and 10% are used for the test. Minibatch size is 100. For the LMO calculation, we set $L = 250$. We used the median trick for the kernel bandwidth. To calculate the MMD, we have to fix the bandwidth of the kernel and we used $h = 2.5$. The results are shown in Figure. 5.6. In Figure. 5.6(a), the vertical axis is the test accuracy and the horizontal axis is wall clock time.



(a) Greedy Monte Carlo

(b) Herding Monte Carlo

(c) Greedy Nelder-Mead

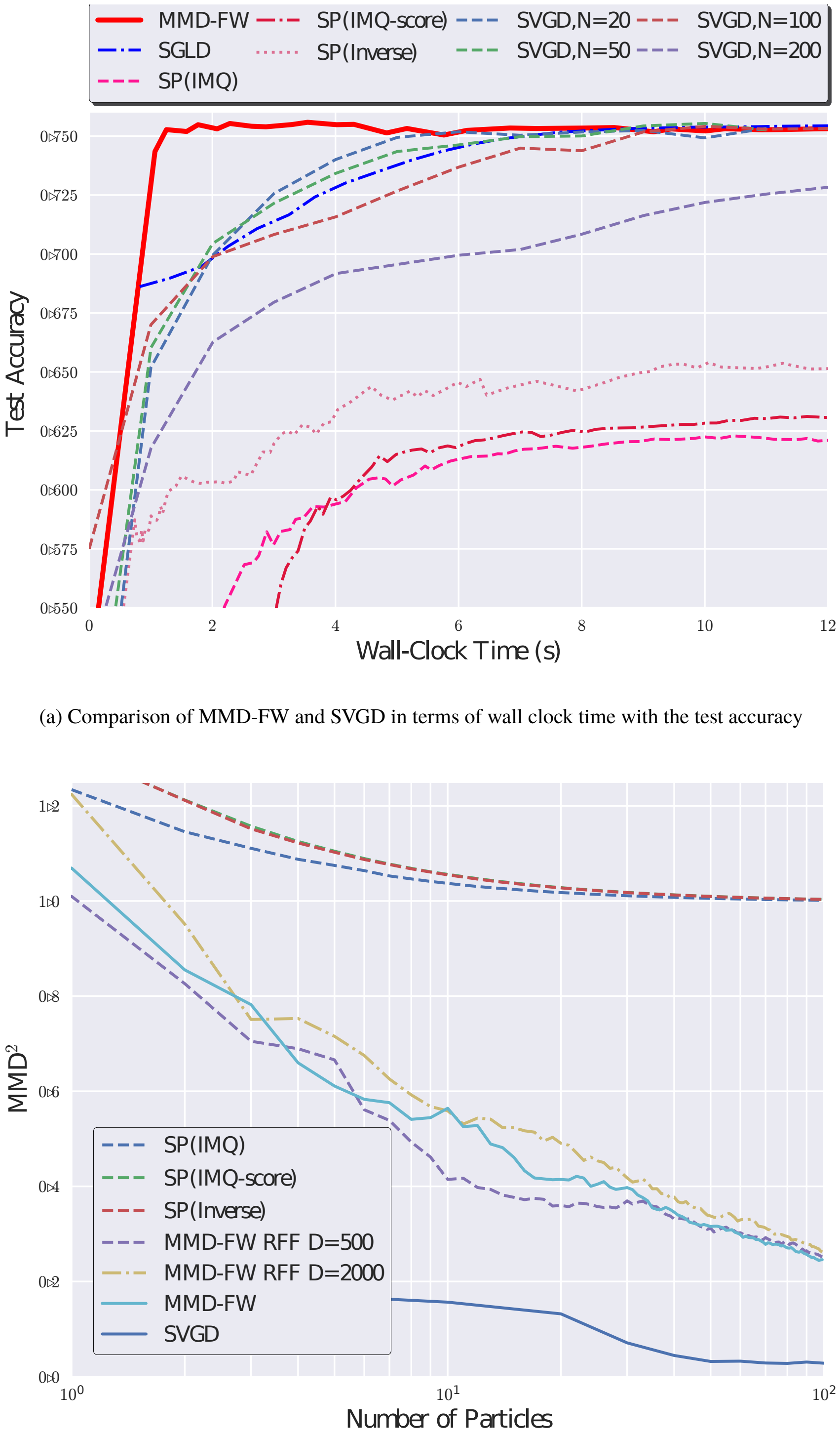
(d) Herding Nelder-Mead

(e) Greedy Grid Search

(f) Herding Grid Search

FIGURE 5.5: Plots of the toy experiments by SPs

As we discussed in Section 5.4.1, our algorithm was faster than SVGD in terms of wall clock time. SP did not work well. We also compared MMD-FW with stochastic gradient Langevin dynamics (SGLD) (Welling and Teh, 2011) and faster than SGLD.



(a) Comparison of MMD-FW and SVGD in terms of wall clock time with the test accuracy

(b) Convergence behavior in terms of number of the particles with MMD^2

FIGURE 5.6: Comparison for the logistic regression model

Figure. 5.6(b) shows the convergence behavior, where the vertical axis is MMD^2 and the horizontal one is the number of particles in the log scale. To calculate MMD, we generated "true samples" by Hamiltonian Monte Carlo (Neal et al., 2011). Since RBF kernel is an infinite dimensional kernel, to further check the convergence behavior under the finite dimensional kernel, we approximated the RBF kernel by random Fourier expansion (RFF). In Figure. 5.6(b), D is the number of frequency of RFF. Also, we still compared with SP on MMD although this comparison is a little unfair since the objective of SP is kernelized Stein discrepancy. As discussed in Section 5.3.2, although the convergence is sub-linear order theoretically since we used RBF kernel which is an infinite dimensional kernel, we observed the linear convergence thanks to the rounding error in the computer. The convergence speed of RBF kernel approximated by RFF showed the linear, which is the expected behavior since the approximated kernel by RFF is the finite dimensional kernel.

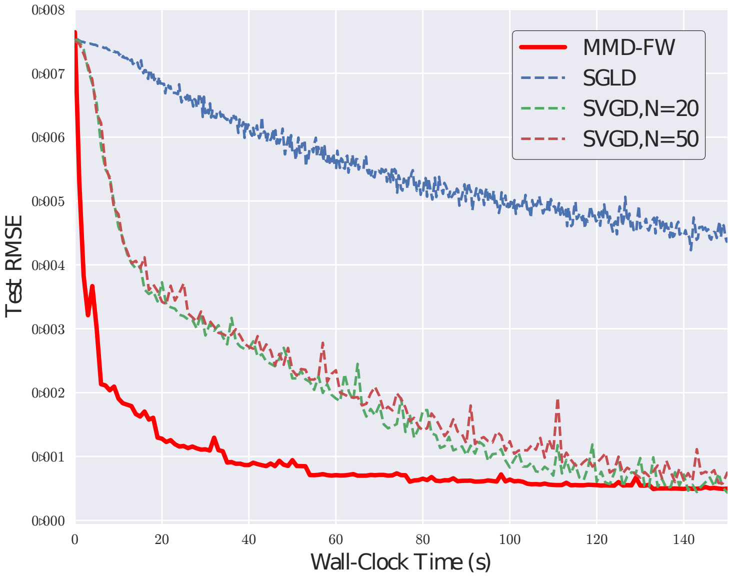
SVGD had a smaller MMD than the proposed method, which is due to the fact that SVGD simultaneously optimizes all particles and tries to put particles in the best position in correspondence with the global optima. In contrast, MMD-FW only increased the particles greedily, and this resulted in local optima. Hence, the better performance of SVGD compared with MMD-FW with the same number of particles in terms of MMD is a natural result.

Bayesian neural net regression

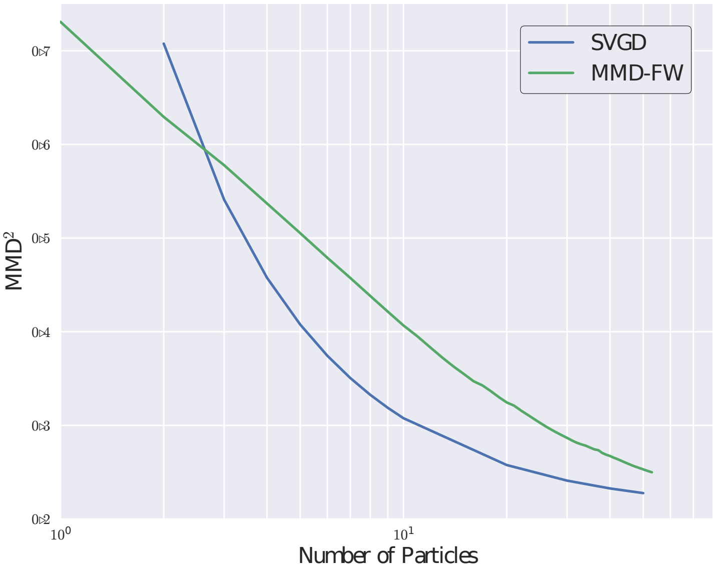
We experimented with Bayesian neural networks for regression. The settings were the same as those in Liu and Wang (2016). We used a neural network with one hidden layer, 50 units, and the ReLU activation function. As the dataset, we used the Naval data from the UCI (Dheeru and Karra Taniskidou, 2017), which contains 11,934 data points and 17 features. In this experiment, we used Adam with a learning rate of 0.005 and we split the data, 90% are used for training and 10% are used for the test. minibatch size is 100 except for year dataset, where we used 500 minibatch sizes. We use the zero mean Gaussian for the prior of the weights and we put $\mathrm{Gamma}(1, 0.1)$ prior for the inverse covariances. For the LMO calculation, we set $L = 1000$ except for year dataset where we set $L = 2000$. The posterior dimension was 953. The results are shown in Figure. 5.7. In Figure. 5.7(a), the vertical axis is the test RMSE, and the horizontal axis is wall clock time. In Fig 5.7(b), the vertical axis is the MMD^2, and the horizontal axis is the number of particles. We show the additional experimental results in Section 5.6.10.The posterior dimension was much higher than that of the logistic regression, but our algorithm was faster than SVGD in terms of wall clock time and linearly converged, which is consistent with the theory.

TABLE 5.1: Benchmark results on test RMSE and log likelihood by Bayesian neural net regression model

Dataset	**Posterior**	Avg. Test RMSE		Avg. Test log likelihood		Fixed Wall clock
	dimension	SVGD	Ours	SVGD	Ours	Time (Secs)
Naval (N=11,934, D=17)	953	4.9e-4±7.5e-5	**4.2e-4±5.3e-5**	6.08 ± 0.11	**6.00±0.12**	150
Protein (N=45730, D=9)	553	4.51 ± 0.057	**4.43±0.035**	-2.93 ± 0.013	**-2.91±0.0073**	40
Year (N=515344, D=91)	9203	9.54 ± 0.08	**9.50±0.09**	**-3.65±0.005**	**-3.65±0.011**	300



(a) Comparison of MMD-FW and SVGD in terms of wall clock time with test RMSE



(b) Convergence behavior in terms of the number of the particles with MMD^2

FIGURE 5.7: Comparison for the Bayesian neural net regression model

Results for other datasets are shown in Table 5.1, where we fixed the wall clock time and applied MMD-FW and SVGD within that period. SP did not work well because of the high dimensionality so its results are not shown. We experimented 5 random trials for changing the splitting of the dataset. For the Protein data, we used the same model as the Naval data, and for the Year data, we used the same model as others except that the number of hidden units is 100. From these benchmark dataset experiments, we confirmed that our method shows almost the same performance as SVGD in many cases but shows faster optimization. Moreover, it shows linear convergence.

5.6 Appendix

In this section, we describe the proofs, additional discussions, and detailed explanations for the experimental settings.

5.6.1 Proof of the gradient Eq.(5.15)

For the symmetric kernel k, the relation $\nabla_\theta k(\theta, \theta') = \nabla_{\theta'} k(\theta', \theta)$ holds, and we apply the partial integral method to the first term, then

$$\begin{aligned}\nabla_{\theta_n} \int k(\theta, \theta_n) p(\theta) d\theta &= \int \nabla_{\theta_n} k(\theta, \theta_n) p(\theta) d\theta = \int \{\nabla_\theta k(\theta_n, \theta)\} p(\theta) d\theta \\ &= k(\theta_n, \theta) p(\theta)\big|_{-\infty}^{\infty} - \int k(\theta_n, \theta) \nabla_\theta p(\theta) d\theta = -\mathbb{E}_{p(\theta)}\left[k(\theta, \theta_n) \nabla_\theta \ln p(\theta)\right]. \end{aligned} \tag{5.19}$$

To approximate the integral, we usually use importance sampling when the analytic form of the integral is not available. Instead, MMD-FW is the greedy approach, therefore we have particles which had already been processed. Thus, we approximate the expectation by the empirical distributions which are composed of already obtained particles. The FW framework does not need the exact solution of the LMO and we just approximately solve it. At the early stage of the algorithm, there are not so many particles and there might exist the unreliable particles, hence the expectation is not so reliable. Since we only need to solve the approx-LMO, we can use these particles. Specifically, the justification of using the existing particle for the integral approximation is based on no need to strictly solve the LMO. Although the gradient is incorrect, the LMO can be solved with error to some extent because the first particle is close to MAP and the evaluation points of the expectation include, at least, one region with high density on $p(\theta)$. If the LMO is δ-close to the true value, the weights of old incorrect particles will be updated to be small enough to ignore as the algorithm proceeds. Therefore the framework using processed particles works. The key trick for this is that the initial particle is close to the MAP. This kind of inexact gradient descent is widely used in the FW algorithm. And as the algorithm proceeds, the weights of those early unreliable particles are gradually reduced by the step size. Thus, we solve the LMO by the gradient descent of which gradient is written by

$$\nabla_{\theta_n} \int k(\theta, \theta_n) p(\theta) d\theta \simeq -\frac{1}{N} \sum_{m=1}^{N} k(\theta_m, \theta_n) \nabla_{\theta_m} \ln p(\theta_m). \tag{5.20}$$

Thus, we can obtain the update equation,

$$\nabla_\theta \langle \mu_{\hat{p}}^{(n)} - \mu_p, g \rangle = \frac{1}{n} \sum_{l=1}^{n} w_l^{(n)} \nabla_\theta k(\theta_l, \theta) + \frac{1}{n} \sum_{l=1}^{n} w_l^{(n)} k(\theta, \theta_l) \nabla_{\theta_l} \ln p(\theta_l). \tag{5.21}$$

Algorithm 5: MMD minimization by Frank-Wolfe algorithm (MMD-FW)

```
Input: A posterior density $p(\theta)$
Output: A set of particles $(\{w_i, \theta_i\}_{i=1}^N)$
Calculate approximate MAP estimation for $\mu_{\hat{p}}^{(1)}$
for $n = 2 \dots N$ do
    $k(\cdot, \theta_n)$ =Approx-LMO($\mu_{\hat{p}}^{(n-1)}$)
    if Constant step then
        $\lambda_n = \frac{1}{\gamma+1}$
        Update $\mu_{\hat{p}}^{(n+1)} = (1-\lambda_l)\mu_{\hat{p}}^{(n)} + \lambda_n \bar{g}_n$
    else if Line search: then
        $\lambda_n = \mathrm{argmin}_{\lambda \in [0,1]} J((1-\lambda)\mu_{\hat{p}}^{(n)} + \lambda \bar{g}_n)$
        Update $\mu_{\hat{p}}^{(n+1)} = (1-\lambda_l)\mu_{\hat{p}}^{(n)} + \lambda_n \bar{g}_n$
    else
        Empirical BQ weight: $\hat{w}_i^n = \sum_{m=1}^n \hat{z}_m K_{im}^{-1}, \hat{z}_m = \sum_{l=1}^n k(\theta_l, \theta_m)/n$
        Update $\mu_{\hat{p}}^{(n+1)} = \sum_{i=1}^n \hat{w}_i^n k(\theta, \theta_i)$
    end if
end for
```

In the above expression, the first term corresponds to the regularization term, which tries to scatter the particles. When we use the RBF kernel, the first term is proportional to the inverse of the bandwidth. Thus, it is easily understood that small bandwidth makes regularization term large, and vise Versa. The second term tries to move particles in high mass regions.

5.6.2 Discussion about the step size of FW

A step size selection is crucial for the success of the FW algorithm since both the empirical performance and theoretical convergence rate strongly depends on the step size. Generally, there are three choices as shown in Alg 5. Common choices of the step sizes are the constant step size and Line search. The step size of line search can be written as

$$\lambda_n = \frac{\langle \mu_{\hat{p}}^{(n)} - \mu_p, \mu_{\hat{p}}^{(n)} - \bar{g}_{n-1} \rangle}{\|g_{i-1} - \bar{g}_n\|_{\mathcal{H}}^2}. \tag{5.22}$$

The point is that they constantly reduce the weights of earlier particles. Thus, those step sizes are preferable when the early particles are not reliable. In our algorithm, those weights are not preferable since we use the near MAP initialization.

Another choice of the step size is the fully correction. As the name means, this method updates the weights of all particles which have already been obtained at the previous steps. The Bayesian quadrature (BQ) weights are categorized into this type. In our algorithm, we used the BQ weights since they are the optimal weights for the MMD. For more details of the Bayesian Quadrature, please see Section 5.6.13 The weights of BQ can be calculated by

$$w_{\mathrm{BQ}}^{(n)} = \sum_m z_j^\top K_{nm}^{-1}, \tag{5.23}$$

where K is the gram matrix, $z_n = \int k(\theta, \theta_n)p(\theta)d\theta$ and we approximate the integral by particles. Fully correction is preferable when the early particles are important. So, this choice is preferable in our algorithm

The step size choice affects the convergence rate directly. In Section 5.3.2, we only showed the results of the BQ weights. Actually, line search error bound is the same as the BQ weights. Also, the infinite RKHS result of constant step is the same sa the BQ weights. Here we show the error bound of constant step and $\mathcal{H}$ is finite dimensional.

Theorem 9. (Consistency) *Under the condition of Theorem 7, the error $|Z_{f,p} - Z_{f,\hat{p}}|$ of Alg. 5 with constant step size is bounded at the following rate:*

$$|Z_{f,p} - Z_{f,\hat{p}}| \leq \mathrm{MMD}(\{(w_n, \theta_n)\}_{n=1}^{N}) \leq \frac{2r^2}{R\delta N}, \tag{5.24}$$

where r is the diameter of the marginal polytope $\mathcal{M}$, δ is the accuracy parameter, and R is the radius of the smallest ball of center μ_p included $\mathcal{M}$.

Also, after MMD-FW algorithm converges, when we reweight the obtained particles by using Bayesian quadrature weights. Then we can interpret it as the posterior, the following contraction property holds (line search result is the same as the BQ, and infinite RKHS result of constant step is the same as BQ, we only show the result of constant step in finite RKHS).

Theorem 10. (Contraction) *Let $S \subseteq \mathbb{R}$ be an open neighborhood of the true integral $Z_{f,p}$ and let $\gamma = \inf_{r \in S^c} |r - Z_{f,p}| > 0$. Then the posterior probability of mass on $S^c = \mathbb{R} \setminus S$ by Alg 5 with constant step size vanishes at the rate:*

$$\mathrm{prob}(S^c) \leq \frac{2\sqrt{2}r^2}{\sqrt{\pi}R\delta\gamma N} e^{-\frac{\gamma^2 R^2 \delta^2 N^2}{8r^4}}, \tag{5.25}$$

where d is the diameter of the marginal polytope $\mathcal{M}$, δ is the accuracy parameter, R is the radius of the smallest ball of center μ_p included $\mathcal{M}$.

5.6.3 Proof of Theorem 7

First, we consider the case of Line search variants. The proof goes almost in the same way as Beck and Teboulle (2004) for the finite dimensional RKHS. (The proof of Guélat and Marcotte (1986) is also useful.)

Finite dimensional RKHS

Since RKHS is finite-dimensional, we can prove the theorem in the same way as

1. First, we rewritten the proof of Proposition 3.2. in Beck and Teboulle (2004) where the proof is done on $\mathbb{R}^n$ and then we extend it to the situation where we use the approximate LMO. The

problem in Beck and Teboulle (2004) is

$$v^* = \min_{h\in\mathcal{S}} \frac{1}{2}\|Mh - g\|^2, \tag{5.26}$$

and we solve this by FW. Here, g is m-dimensional function and h is the n-dimensional function, and M is the projection matrix. And $\mathcal{S}$ denotes the feasible space of functions that we are considering, such as RKHS. First, we state the general strategy of proving the linear convergence of the above problem by the line search FW algorithm. If you want to see the whole proof, please check Beck and Teboulle (2004).

We consider to solve the above problem by FW algorithm. We express the solution of the linearization of the above problem as p, that is, if we express the initial point as $h_0 \in \mathcal{M}$, and express the $k-1$-th linearization problem and its solution as $p_{k-1} := \arg\min_{p\in\mathcal{S}}\{\langle p - h_{k-1}, \nabla f(h_{k-1})\rangle\}$. And if the step size λ_{k-1} is obtained via constant step or Line search, then the next state is calculated by $h_k = h_{k-1}+\lambda_{k-1}(p_{k-1}-h_{k-1})$. We express $v_k := g-Mh_{k-1}$, $w_k := g - Mp_{k-1}$. Base on this definition, $\nabla f(h_{k-1}) = M^\top(Mh_{k-1} - g)$, thus LMO problem can be written as

$$\begin{aligned}
p_{k-1} :&= \arg\min_{p\in\mathcal{S}}\{\langle p - h_{k-1}, M^\top(Mh_{k-1} - g)\rangle\} \\
&= \arg\min_{p\in\mathcal{S}}\{\langle M(p - h_{k-1}) - g + g, Mh_{k-1} - g\rangle\} \\
&= \arg\min_{p\in\mathcal{S}}\{\langle Mp - g + v_{k-1}, -v_{k-1}\rangle\} \\
&= \arg\min_{p\in\mathcal{S}}\{\langle g - Mp, v_{k-1}\rangle\}.
\end{aligned} \tag{5.27}$$

Thus, the LMO problem can be characterized as

$$\langle w_{k-1}, v_{k-1}\rangle = \min_{p\in\mathcal{S}}\{\langle g - Mp, v_{k-1}\rangle\} \tag{5.28}$$

Also $\|v_k\|^2$ denotes the error of the algorithm at k-th step.

Let us consider the line search step size. By the straightforward calculation of the definition of the line search, we can show that the line search step size is

$$\lambda_{k-1} = \frac{\langle v_{k-1}, v_{k-1} - w_{k-1}\rangle}{\|v_{k-1} - w_{k-1}\|^2}. \tag{5.29}$$

if this $\lambda_{k-1} \leq 1$, since we assumed that the step size is smaller than 1. Based on this step size, we can show that

$$\|v_k\|^2 = \|g - Mh_k\|^2 = \frac{\|v_{k-1}\|^2\|w_{k-1}\|^2 - \langle v_{k-1}, w_{k-1}\rangle^2}{\|v_{k-1} - w_{k-1}\|^2}. \tag{5.30}$$

From Proposition 3.1 in Beck and Teboulle (2004), following relation holds,

$$\langle v_k, w_k \rangle \leq -R_s(\hat{h}, M)\|v_k\|. \tag{5.31}$$

This says that there exists a ball whose radius is $R_s(\hat{h}, M)$ centerted lies within $\mathcal{M}$. By using this relation, we can show that

$$\|v_k\|^2 \leq \left(1 - \frac{R^2}{\|w_{k-1}\|^2}\right)\|v_{k-1}\|^2. \tag{5.32}$$

Finally, since the domain $\mathcal{S}$ is the bounded set, it is contained in some larger ball whose radius is ρ_S and thus, the relation $\|w_{k-1}\| \leq \|g - Mp_{k-1}\| \leq \|g\| + \|M\|\rho_s$ holds. Thus

$$\|v_k\|^2 \leq \left(1 - \left(\frac{R}{\|g\| + M\|\rho_s\|}\right)^2\right)\|v_{k-1}\|^2, \tag{5.33}$$

holds and this means the linear convergence of the problem, since $\|v_k\|$ express the error of the algorithm at iteration k.

2. Base on the original proof, let us consider the approximate LMO whose accuracy parameter is δ. As we saw, the solution of the LMO problem can be written as Eq.(5.28). Approximate LMO returns $\tilde{w}$ which deviates from the true w in the following way Also $\|v_k\|^2$ denotes the error of the algorithm at k-th step.

   Let us consider the line search step size. By the straightforward calculation of the definition of the line search, we can show that the line search step size is

$$\langle v_k, \tilde{w}_k \rangle \leq \delta\langle v_k, w_k \rangle. \tag{5.34}$$

   This is derived straightforwardly from the definition of the approximate LMO. From this definition, following holds by Eq.(5.31)

$$\langle v_k, \tilde{w}_k \rangle \leq -\delta R_s(\hat{h}, M)\|v_k\|. \tag{5.35}$$

   Here after, for simplicity, we assume that step size of the line search and BQ are obtained without approximation(In our algorithm, they are approximated by empirical approximation). Later, we will discuss the those inexact step sizes.

   Based on the above approximate LMO relation, we replace the w_k by $\tilde{w}_k$ in the proof of Proposition 3.2. in Beck and Teboulle (2004), and we obtain the variant of Eq.(12) in Beck and Teboulle (2004) which uses approximate LMO not LMO. After this, we use Eq.(5.35) for the evaluation of $\|v_k\|^2$ and we can obtain the following expression,

$$\|v_k\|^2 \leq \left(1 - \left(\frac{\delta R_s(\hat{h}, M)}{\|g\| + \rho_s\|M\|}\right)^2\right)\|v_{k-1}\|^2 \tag{5.36}$$

Finally let us rewrite Eq.(5.36) by using the notations which we used in Section 5.3. From the assumption that $\|k(\cdot,\theta)\|_{\mathcal{H}} \leq r$, this means that r is the diameter of the marginal polytope. Thus $\|g\| + \rho_s\|M\| \leq r$ and $\|v_0\|^2 = \|g - Mh_0\|^2 \leq r^2$. Thus,

$$\|v_k\|^2 \leq r^2 \exp\left(-N\left(\frac{\delta R}{r}\right)^2\right). \tag{5.37}$$

Based on this bound, we can apply the result of Ch.4.2 in Bach et al. (2012).

Then, by utilizing the discussion of Section B in Briol et al. (2015), we can obtain the following expression.

$$|Z_{f,p} - Z_{f,\hat{p}}| \leq \mathrm{MMD}(\{(w_n, h_n)\}_{n=1}^N) \leq \|\mu_p - \mu_{\hat{p}}\|. \tag{5.38}$$

This is derived Cauchy Schwartz inequality and the definition of MMD and $\|f\|_{\mathcal{H}} \leq 1$. Thus, we have proved the theorem in the case of line search.

3. Since fully corrective variants optimize all the weights, the upper bound of this is superior to that of the line search (In particular, BQ weights are the optimal weights). Hence

$$|Z_{f,p} - Z_{f,\hat{p}}|^2 \leq \|v_k^{\mathrm{FC}}\|^2 \leq \|v_k\|^2 \leq r^2 \exp\left(-N\left(\frac{\delta R}{r}\right)^2\right), \tag{5.39}$$

where v_k^{FC} is derived by fully corrective variants. Thus we can bound the fully corrective variant in the same expression as line search. Also, geometric convergence of fully correction variant is discussed in Locatello et al. (2017a); Lacoste-Julien and Jaggi (2015). They also discussed it by using the fact that fully correction is superior to line search. So far we have worked on the problem in Beck and Teboulle (2004), but this result is directly applicable to the finite-dimensional RKHS problem, see Bach et al. (2012).

This is the result when the exact step size is available. In Section 5.6.5.2, we will consider the effect of inexact step size (and introduce δ_{BQ})

Finite dimensional RKHS and constant step size case

Next, we consider the constant step case. This proof is completely same as Chen et al. (2010); Bach et al. (2012) except for replacing the LMO to approx-LMO and introduce the accuracy parameter δ.

Different accuracy of approx-LMO

In the above proof, we consider the fixed accuracy parameter δ for the approximate LMO. However, the δ can be different at each approximate LMO calls. In that situation, we express the accuracy parameter of the k-th call as δ_k. We consider the worst accuracy LMO call and define $\delta = \min_k \delta_k$.

About the Line search, if we put $q^2 = \left(\frac{R_s(\hat{\theta},M)}{\|g\|+\rho_s\|M\|}\right)^2$, then following relation holds,

$$\begin{aligned}
\|v_k\|^2 &\leq \|v_0\|^2 e^{-q^2\sum_{l=0}^{k}\delta_k} \\
&\leq \|v_0\|^2 e^{-q^2 k \min_k \delta_k} \\
&= \|v_0\|^2 e^{-q^2 k\delta}.
\end{aligned} \tag{5.40}$$

Infinite dimensional RKHS

The above discussion depends on the existence of a ball inside the domain. Next we discuss about the infinite RKHS situation, where a ball does not exist. For the proof, we just utilize the standard FW proof. The proof is the same in Locatello et al. (2017b). We use the same notation in the finite RKHS case. We define the objective function as $f(h) = \frac{1}{2}\|h-\mu\|^2$. This norm should be written as $\|\cdot - \cdot\|_{\mathcal{H}}$, but for simplicity, we write in the above form. Let us assume that after the k-th step of FW, we get the solution of approx-LMO and which is expressed by p. Then, the solution h_{k-1} is updated to $h_k = h_{k-1} + \lambda(p - h_{k-1})$ where λ is the step size. We study how $f(h_k)$ and $f(h_{k-1})$ is different. We expand $f(h_k)$ as

$$f(h+\lambda(p-h)) = f(h) + \lambda\langle p-h, \nabla f(h)\rangle + \frac{\lambda^2}{2}\|(h-p)\|^2. \tag{5.41}$$

About the third term, from the assumption that $\|k(\cdot,\theta)\|_{\mathcal{H}} \leq r$, thus this means that r is the diameter of the marginal polytope it is upper bounded by $\frac{\lambda^2}{2}(2r)^2$. Then we set $v_k = \nabla f(h_k)$, then $\|v_k\|^2/2 = f(h_k)$. About the second term, by using this definition and the approx-LMO, $\langle p_{k-1} - h_{k-1}, \nabla f(h_{k-1})\rangle \leq -\delta\|v_{k-1}\|^2$.

So far we do not specify the step size. Here let us assume that Line search step size is used. Then line search step size minimizes the right hand side of the above inequality with respect to λ. From this, by using the approx-LMO with accuracy parameter δ, we get the following inequality

$$\begin{aligned}
\|v_{k+1}\|^2 &\leq \|v_k\|^2 + \min_{\lambda}\left\{-\lambda\delta\|v_k\|^2 + \frac{\lambda^2}{2}(2r)^2\right\} \\
&\leq \|v_k\|^2 - \frac{2}{\delta k+2}\delta\|v_k\|^2 + \frac{2}{(\delta k+2)^2}(2r)^2,
\end{aligned} \tag{5.42}$$

where in the second line, we set $\lambda = \frac{2}{\delta k+2}$. This step size is not optimal, on the other hand line search step size is optimal, thus we get the inequality in the second line on the above expression.

Finally, by the induction, we get prove

$$\|v_k\|^2 \leq 2\frac{(1+\delta)(2r)^2}{\delta(\delta k+2)}, \tag{5.43}$$

this is the same way as the standard FW algorithm. This ends the proof when exact step size calculation is available. In Section 5.6.5.2, we will consider the effect of inexact step size (and introduce δ_{BQ}).

5.6.4 Discussion about the inexact step size for line search step size

Here, we analyze the effect of inexact step size on the convergence rate.

First, we will see the step size of the line search in finite-dimensional case. The calculation of the step size in the line search includes $\langle \mu_p, g \rangle = \int k(\theta, \theta')p(\theta')d\theta'$ which is intractable in general if $p(\theta)$ is posterior distribution. For the analysis, we express the exactly calculated step size by the line search by λ and λ' denotes the step size in which the above integration is approximated by empirical distribution. We also express the ratio of λ and λ' as α, where $\lambda' = \alpha\lambda$. This α express the deviation from the exact step size λ. We analyze what range of α is required to assure the linear convergence in finite dimensional kernel

Theorem 11. *(Inexact step size in for line search) If the ratio α is bounded inside $(0, 2)$, then exponential convergence still holds.*

Proof. First, let us go back to the proof of exponential convergence for line search step size. Since we express the approximated step size by $\lambda' = \alpha\lambda$,

$$\begin{aligned}
&\|v_{k+1}^2\| \\
&= \|g - Mh_{k+1}\|^2 \\
&= \lambda'^2\|v_k - w_k\|^2 + 2\lambda'\langle v_k, w_k - v_k\rangle + \|v_k\|^2 \\
&= (1-\alpha)^2\|v_k\|^2 - 2(1-\alpha)^2\langle v_k, w_k\rangle + (\alpha^2 - 2\alpha)\langle v_k, w_k\rangle^2 + \|v_k\|^2\|w_k\|^2.
\end{aligned} \tag{5.44}$$

From this, we can bound the right hand side in the same way as before, but which includes the additional coefficients

$$\|v_{k+1}^2\| \leq \left\{1 - \frac{\alpha(2-\alpha)R^2}{r^2}\right\}\|v_k^2\|. \tag{5.45}$$

Thus, to enhance the geometrical decrease, $\alpha(2-\alpha) > 0$ is needed. This ends the proof. Note that the convergence rate is maximized at $\alpha = 1$, that is, the correct step size is used. □

5.6.5 Discussion about BQ weights and proof of Theorem 6

Next, we analyze the approximate BQ weights. To do that, we need to evaluate how MMD changes between the $k+1$-th step of FW and k-th step of FW. To evaluate this difference, we first review the Bayesian Quadrature (BQ).

5.6.5.1 Discussion about Bayesian quadrature and inexact step size

In the Bayesian Quadrature method (Ghahramani and Rasmussen, 2003; Huszár and Duvenaud, 2012), we put on the Gaussian process prior on f with kernel k and mean 0. In usual Gaussian processes, after conditioned on $f(\Theta) = (f(\theta_1), \ldots, f(\theta_N))^\top$, we can obtain the closed-form posterior distribution of f,

$$p(f(\theta_*)|p(f(\theta))) = N(f(\theta_*)|\mu, \Sigma), \tag{5.46}$$

where $\mu = k(\theta_*, \theta)K^{-1}f(\Theta)$, $\Sigma = k(\theta_*, \theta_*) - k(\theta_*, \theta)K^{-1}k(\theta, \theta_*)$, here $K_{i,j} = k(\theta_i, \theta_j)$ and $N(x|\mu, \Sigma)$ means the Gaussian distribution with mean μ and the covariance Σ. Thanks to the property of Gaussian process that linear projection preserves the normality, the integrand is also Gaussian, and thus we can obtain the posterior distribution of the integrand as follows,

$$\begin{aligned}
\mathbb{E}_{\mathrm{GP}}[Z_{f,p}] &= \mathbb{E}_{\mathrm{GP}}\left[\int f(\theta)p(\theta)d\theta\right] \\
&= \iint f(\theta)p\left(f(\theta)|p(f(\Theta))\right)p(\theta)d\theta df \\
&= \int k(\theta, \Theta)K^{-1}f(\Theta)p(\theta)d\theta \\
&= \boldsymbol{z}^\top K^{-1}f(\Theta),
\end{aligned} \tag{5.47}$$

where $z_n = \int k(\theta, \theta_n)p(\theta)d\theta$. From the above expression,

$$\mathbb{E}_{\mathrm{GP}}[Z_{f,p}] = \sum_{n=1}^{N} w_{\mathrm{BQ}}^{(n)}f(\theta_n), \;\; w_{\mathrm{BQ}}^{(n)} = \sum_m z_j^\top K_{nm}^{-1}. \tag{5.48}$$

In the same way as the expectation, we can calculate the variance of the posterior,

$$\mathbb{V}[Z_{f,p}|f(\theta_1), \ldots f(\theta_N)] = \iint k(\theta, \theta')p(\theta)p(\theta')d\theta d\theta' - \boldsymbol{z}^\top K^{-1}\boldsymbol{z}. \tag{5.49}$$

Huszár and Duvenaud (2012) proved that in the RKHS setting, minimizing the posterior variance corresponds to minimizing the MMD,

$$\mathbb{V}[Z_{f,p}|f(\theta_1), \ldots f(\theta_N)] = \mathrm{MMD}^2(\{(w_{\mathrm{BQ}}^{(n)}, \theta_n)\}_{n=1}^N). \tag{5.50}$$

The BQ minimize the above discrepancy greedily in the following way,

$$\theta_{N+1} \leftarrow \arg\min_\theta \mathbb{V}[Z_{f,p}|f(\theta_1), \ldots f(\theta_N), f(\theta)]. \tag{5.51}$$

Huszár and Duvenaud (2012) showed that

$$\mathrm{MMD}(\{(w_{\mathrm{BQ}}^{(n)}, \theta_n)\}_{n=1}^N) = \inf_{w \in \mathbb{R}^N} \sup_{f \in \mathcal{H}: \|f\|_{\mathcal{H}}=1} |Z_{f,p} - \hat{Z}_{f,p}|, \tag{5.52}$$

and thus,

$$\mathrm{MMD}(\{(w_{\mathrm{BQ}}^{(n)}, \theta_n)\}_{n=1}^N) \leq \mathrm{MMD}(\{(w_n = \frac{1}{N}, \theta_n)\}_{n=1}^N) \tag{5.53}$$

Now we analyze how $\mathrm{MMD}(\{(w_{\mathrm{BQ}}^{(n)},\theta_n)\}_{n=1}^{k+1})^2$ and $\mathrm{MMD}(\{(w_{\mathrm{BQ}}^{(n)},\theta_n)\}_{n=1}^{k})^2$ differs. This is explicitly calculated by Eq.(5.49),

$$\begin{aligned}&\mathrm{MMD}(\{(w_{\mathrm{BQ}}^{(n)},\theta_n)\}_{n=1}^{k+1})^2-\mathrm{MMD}(\{(w_{\mathrm{BQ}}^{(n)},\theta_n)\}_{n=1}^{k})^2\\&\quad=-\boldsymbol{z}_{(k+1)}^\top K_{(k+1)}^{-1}\boldsymbol{z}_{(k+1)}+\boldsymbol{z}_{(k)}^\top K_{(k)}^{-1}\boldsymbol{z}_{(k)},\end{aligned} \tag{5.54}$$

where $K_{(k)}$ denotes the Gram matrix using data θ_1 to θ_k and $\boldsymbol{z}_{(k)}=(\int k(\theta_1,\theta)d\theta,\dots,\int k(\theta_k,\theta)d\theta)^\top$. Since this quantity is the difference of quadratic form, it is convenient for the analysis based on their eigenvalues. Here we assume that $K_{(k)}$ and $K_{(k+1)}$ are full rank. Since they are gram matrix of positive definite kernel, there exists different positive k eigenvalues for the matrix $K_{(k)}$. We denote those eigenvalues by $\gamma_i, i=1\dots k$, and let e_i be its eigenvector, $K_{(k)}e_i=\gamma e_i$. Let $U=(e_1,\dots,e_k)$, then by diagonalization

$$\begin{aligned}K_{(k)}&=U\begin{pmatrix}\gamma_1&\dots&0\\&\ddots&\\0&\dots&\gamma_k\end{pmatrix}U^\top &(5.55)\\&=U\Gamma U^\top. &(5.56)\end{aligned}$$

From the inverse matrix property,

$$\begin{aligned}K_{(k)}^{-1}&=U\begin{pmatrix}\gamma_1^{-1}&\dots&0\\&\ddots&\\0&\dots&\gamma_k^{-1}\end{pmatrix}U^\top &(5.57)\\&=U\Gamma^{-1}U^\top. &(5.58)\end{aligned}$$

By diagonalization,

$$\mathrm{MMD}(\{(w_{\mathrm{BQ}}^{(n)},\theta_n)\}_{n=1}^{k})^2=\sum_{i=1}^{k}\gamma_i^{-1}z_i'^\top z_i', \tag{5.59}$$

where $z_i'=U^\top z_i$. Next, about $K_{(k+1)}$, we investigate its eigenvalues. We can express $K_{(k+1)}$ as

$$K_{(k+1)}=\begin{pmatrix}K_{(k)}&\tilde{k}_{(k+1)}\\\tilde{k}_{(k+1)}^\top&1\end{pmatrix}, \tag{5.60}$$

where $\tilde{k}_{k+1}^\top=(k(\theta_{k+1},x_1)\dots k(\theta_{k+1},\theta_k))^\top$. Let E_k be the $k\times k$ identity matrix. Then the eigenvalue of $K_{(k+1)}$ can be calculated by solving the following equation.

$$0=\det\begin{pmatrix}K_{(k)}-\gamma^*E_k&\tilde{k}_{(k+1)}\\\tilde{k}_{(k+1)}^\top&1-\gamma^*\end{pmatrix}. \tag{5.61}$$

We solve the above equation with respect to γ^* and the obtained γ^* correspond to the eigenvalues. We use the determinant formula,

$$\det \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \det A \,\det(D - CA^{-1}B), \tag{5.62}$$

where regularity is assumed for A and D. Then,

$$0 = \det(K_{(k)} - \gamma^* E_k)\left((1-\gamma^*) - \tilde{k}_{n+1}^\top (K_{(k)} - \gamma^* E_k)^{-1}\tilde{k}_{n+1}\right). \tag{5.63}$$

From the first term, we can see that $K_{(k+1)}$ have $(\gamma_1, \ldots, \gamma_k)$ as the eigenvalues. This is equivalent to the eigenvalue of K_n. The newly appearing eigenvalue is the solution of

$$0 = (1-\gamma^*) - \tilde{k}_{(k+1)}^\top (K_{(k)} - \gamma^* E_k)^{-1}\tilde{k}_{(k+1)}. \tag{5.64}$$

This eigenvalue is also positive and it is different from other eigenvalues $(\gamma_1, \ldots, \gamma_k)$. We express the solution of the above equation as γ_{k+1}. Let us goes back to the evaluation of the difference of MMD between $k+1$ and k-th step of FW,

$$\mathrm{MMD}(\{(w_{\mathrm{BQ}}^{(n)}, \theta_n)\}_{n=1}^{k+1})^2 - \mathrm{MMD}(\{(w_{\mathrm{BQ}}^{(n)}, \theta_n)\}_{n=1}^{k})^2. \tag{5.65}$$

For the evaluation of the above difference, in addition to the eigenvalue, we also need the eigenvector of the gram matrix $K_{(k+1)}$. From Eq.(5.60), we expand this matrix. For simplicity, we express $\mathbf{a} = \tilde{k}_{(k+1)}$. Then we can express $K_{(k+1)}$ as,

$$K_{(k+1)} = Q^\top \begin{pmatrix} K_{(k)} & 0 \\ 0 & 1 - \mathbf{a}^\top K_{(k)}^{-1}\mathbf{a} \end{pmatrix} Q, \tag{5.66}$$

where

$$Q = \begin{pmatrix} E_k & K_{(k)}^{-1}\mathbf{a} \\ 0 & 1 \end{pmatrix}. \tag{5.67}$$

This is tricky but you can easily verify it by just substituting the definitions. Here we consider the following where $d = 1 - \mathbf{a}^\top K_{(k)}^{-1}\mathbf{a}$ for simplicity,

$$\begin{pmatrix} K_{(k)} & 0 \\ 0 & d \end{pmatrix}\begin{pmatrix} e_i \\ 0 \end{pmatrix} = \begin{pmatrix} K_{(k)}e_i \\ 0 \end{pmatrix} = \lambda_i \begin{pmatrix} e_i \\ 0 \end{pmatrix} := \lambda_i e_i', \tag{5.68}$$

where $(e_i, \lambda_i)_{i=1}^k$ are the eigenvectors and eigenvalues for $K_{(k)}$. Thus, this e_i' can be regarded as the eigenvector for $K_{(k+1)}$ whose eigenvalue is λ_i. Also, by noticing the fact that

$$\begin{pmatrix} K_{(k)} & 0 \\ 0 & d \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = d \begin{pmatrix} 0 \\ 1 \end{pmatrix} =: d e_{k+1}. \tag{5.69}$$

This is also the eigenvector whose eigenvalue is d, also this is orthogonal to $e'_1, \ldots, e'_k$. By setting $U' = (e'_1, \ldots, e'_k, e_{k+1})$, we can diagonalize $K_{(k+1)}$ as

$$K_{(k+1)} = Q^\top U' \begin{pmatrix} \gamma_1 & \ldots & & 0 \\ & \ddots & & \\ & & \gamma_k & \\ 0 & \ldots & & d \end{pmatrix} U'^\top Q \tag{5.70}$$

$$= Q^\top U' \Gamma' U'^\top Q. \tag{5.71}$$

Thus,

$$K^{-1}_{(k+1)} = Q^\top U' \Gamma'^{-1} U'^\top Q. \tag{5.72}$$

Let us calculate $U'^\top Q$ furthur,

$$U'^\top Q = \begin{pmatrix} U^\top & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} E_k & K^{-1}_{(k)}\mathbf{a} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} U^\top & U^\top K^{-1}_{(k)}\mathbf{a} \\ 0 & 1 \end{pmatrix}. \tag{5.73}$$

Let us multiply $(z_1, \ldots, z_k, z_{k+1})^\top = (\mathbf{z}, z_{k+1})^\top$ to the above expression,

$$U'^\top Q \begin{pmatrix} \mathbf{z} \\ z_{k+1} \end{pmatrix} = \begin{pmatrix} U^\top & U^\top K^{-1}_{(k)}\mathbf{a} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{z} \\ z_{k+1} \end{pmatrix}$$
$$= \begin{pmatrix} U^\top \mathbf{z} \\ 0 \end{pmatrix} + z_{k+1} \begin{pmatrix} U^\top K^{-1}_{(k)}\mathbf{a} \\ 1 \end{pmatrix}$$

Based on these results, let us calculate how the variance changes when we add the one data in BQ. Let us go back to Eq.(5.54),

$$\mathrm{MMD}(\{(w^{(n)}_{\mathrm{BQ}}, \theta_n)\}^{k+1}_{n=1})^2 - \mathrm{MMD}(\{(w^{(n)}_{\mathrm{BQ}}, \theta_n)\}^{k}_{n=1})^2$$
$$= -z^2_{k+1} \left((U^\top K^{-1}_{(k)}\mathbf{a})^\top, 1 \right) \Gamma^{-1'} \begin{pmatrix} U^\top K^{-1}_{(k)}\mathbf{a} \\ 1 \end{pmatrix}$$
$$= -\alpha z^2_{k+1} < 0. \tag{5.74}$$

Thus, we confirm that how much each step of the FW algorithm decreases MMD.

5.6.5.2 Proof of Theorem 6

Based on these results, let us describe the proof of Theorem 6 for the inexact step sizes. For the notation, let us denote $\mathrm{MMD}(\{(w^{(n)}_{\mathrm{BQ}}, \theta_n)\}^{k+1}_{n=1})^2$ as $\|v_{k+1}\|^2$. Then we analyze how the convergence rate is affected by the inexact step size. To do that, from Eq.(5.49, 5.50, 5.74), we check the ratio of the variance between k-th and $k+1$-th step as

$$\frac{\|v_{k+1}\|^2}{\|v_k\|^2} = 1 - \frac{\alpha z_{k+1}^2}{\mathbb{E}_{\theta,\theta'\sim p(\theta)}k(\theta,\theta') - \sum_{i=1}^{k}\gamma_i^{-1} z_i'^{\top} z_i'}. \tag{5.75}$$

Remember that this is the similar expression in the proof of the line searh FW. Since the convergence speed of BQ is at least faster than the line search, the convergence coefficient of BQ is larger than that of the line search, we can say that

$$\frac{R^2}{(\|g\| + \rho_s\|M\|)^2} \leq \frac{\alpha z_{k+1}^2}{\mathbb{E}_{\theta,\theta'\sim p(\theta)}k(\theta,\theta') - \sum_{i=1}^{k}\gamma_i^{-1} z_i'^{\top} z_i'}. \tag{5.76}$$

Now let us consider the empirical approximation effect. We express it via the ratio

$$\beta_i = \sum_l k(\theta_i, \theta_l) / \int k(\theta_i, \theta')p(\theta')d\theta, \tag{5.77}$$

where $\tilde{z}_i = \sum_l k(\theta_i, \theta_l)$ and $z_i = \int k(\theta_i, \theta')p(\theta')d\theta$. This is the ratio between exact weight and empirical approximation. Thus,

$$\tilde{z}_i = \beta_i z_i. \tag{5.78}$$

Then if we use the approximate BQ step, the approximated $\frac{\|\tilde{v}_{k+1}\|^2}{\|\tilde{v}_k\|^2}$ (we stress that those $\tilde{v}$s are the variance which is calculated based on the approximated BQ weights) can be written as

$$\begin{aligned}
&\frac{\|\tilde{v}_{k+1}\|^2}{\|\tilde{v}_k\|^2} \\
&= 1 - \frac{\beta_{k+1}^2 \alpha z_{k+1}^2}{\mathbb{E}_{\theta,\theta'\sim p(\theta)}k(\theta,\theta') - \sum_{i,j=0}^{n}\beta_j z_j K_{ij}^{-1}\beta_i z_i} \\
&= 1 - \frac{\alpha z_{k+1}^2}{\mathbb{E}_{\theta,\theta'\sim p(\theta)}k(\theta,\theta') - \sum_{i=1}^{k}\gamma_i^{-1} z_i'^{\top} z_i'} \frac{\beta_{k+1}^2(\mathbb{E}_{\theta,\theta'\sim p(\theta)}k(\theta,\theta') - \sum_{i=1}^{k}\gamma_i^{-1} z_i'^{\top} z_i')}{\mathbb{E}_{\theta,\theta'\sim p(\theta)}k(\theta,\theta') - \sum_{i,j=0}^{n}\beta_j z_j K_{ij}^{-1}\beta_i z_i}.
\end{aligned} \tag{5.79}$$

For the geometric convergence, $\mathbb{E}_{\theta,\theta'\sim p(\theta)}k(\theta,\theta') - \sum_{i,j=0}^{n}\beta_j z_j K_{ij}^{-1}\beta_i z_i$ must be positive. (Since $\mathbb{E}_{\theta,\theta'\sim p(\theta)}k(\theta,\theta') - \sum_{i=1}^{k}\gamma_i^{-1} z_i'^{\top} z_i'$ is always positive.) If this condition is satisfied then we express

$$\delta_{\mathrm{BQ}} = \frac{\beta_{k+1}^2(\mathbb{E}_{\theta,\theta'\sim p(\theta)}k(\theta,\theta') - \sum_{i=1}^{k}\gamma_i^{-1} z_i'^{\top} z_i')}{\mathbb{E}_{\theta,\theta'\sim p(\theta)}k(\theta,\theta') - \sum_{i,j=0}^{n}\beta_j z_j K_{ij}^{-1}\beta_i z_i}. \tag{5.80}$$

which is some positive constant. Then

$$\frac{\delta_{\mathrm{BQ}} R^2}{(\|g\| + \rho_s\|M\|)^2} \leq \frac{\delta_{\mathrm{BQ}}\alpha z_{k+1}^2}{\mathbb{E}_{\theta,\theta'\sim p(\theta)}k(\theta,\theta') - \sum_{i=1}^{k}\gamma_i^{-1} z_i'^{\top} z_i'}, \tag{5.81}$$

holds. This ends the proof of Theorem 6 and Theorem 7 of the case of finite dimension.

Infinite dimensional RKHS

Next we consider the infinite dimensional RKHS. By using the above notation, when we use the inexact BQ step size,

$$\hat{\Delta}_{BQ} := \|\tilde{v}_{k+1}\|^2 - \|\tilde{v}_k\|^2 = -\beta_{k+1}^2 \alpha z_{k+1}^2. \tag{5.82}$$

In the same way, we express the above quantity under exactly calculated BQ step as

$$\Delta_{BQ} := \|v_{k+1}\|^2 - \|v_k\|^2 = -\alpha z_{k+1}^2. \tag{5.83}$$

Let us goes back to the discussion of the previous section. From Eq.(5.42), we get

$$\Delta_{BQ} \leq \min_{\gamma} \left\{ -\gamma\delta\|v_k\|^2 + \frac{\gamma^2}{2}(2r)^2 \right\}. \tag{5.84}$$

Then, following relation holds,

$$\hat{\Delta}_{BQ} \leq \beta_{k+1}^2 \min_{\gamma} \left\{ -\gamma\delta\|v_k\|^2 + \frac{\gamma^2}{2}(2r)^2 \right\}. \tag{5.85}$$

To eliminate the dependence of k form β_{k+1}, let β' is the largest of $(\beta_1, \ldots, \beta_{k+1})$. And in the same way as the previous discussion, we can conclude by the induction that

$$\|v_k\|^2 \leq 2\frac{(1+\beta'^2\delta)(2r)^2}{\delta(\beta'^2\delta k + 2)}. \tag{5.86}$$

If we set $\delta_{\mathrm{BQ}} = \beta'^2$, this ends the proof.

5.6.6 Proof of Theorem 8

Our results are directly obtained by Section B of Briol et al. (2015), which is the proof of the contraction theorem. The calculations after Eq.(26) and Eq.(31) in Briol et al. (2015) holds in our settings. Thus all we need to do is to substitute the variance of ours into Eq.(31) in Briol et al. (2015). In Briol et al. (2015), author proved that since the posterior distribution is the Gaussian distribution $N(Z_{f,\hat{p}}, \sigma_N^2)$ where

$$\sigma_N = \mathrm{MMD}(\{\theta_i, w_i^{\mathrm{BQ}}\}_{i=1}^N), \tag{5.87}$$

and w_i^{BQ} is the Bayesian Quadrature weight. Then posterior probability mass on S^c is calculated by

$$M_N = \int_{S^c} N(Z_{f,\hat{p}}, \sigma_N^2), \tag{5.88}$$

and this value is approximated by the following (this is the Eq.(31) in Briol et al. (2015))

$$M_N \leq \sqrt{\frac{w\sigma_N^2}{\pi\gamma^2}} \exp(-\gamma^2/(2\sigma_N^2)). \tag{5.89}$$

Our variance is derived by reweighting the particles obtained by MMD-FW with Bayesian Quadrature weight and calculate the weighted MMD. This is upper bounded by the bound of the result Theorem 7 because MMD with Bayesian quadrature weights is optimal (see Section 5.6.5). Thus, by substituting the result of Theorem 7 into Eq.(31) in Briol et al. (2015), we obtain the result.

Actually, we cannot calculate the Bayesian Quadrature weight analytically, so we approximate it by obtained particles. Even in such a case, we can obtain the upper bound. The posterior distribution is denoted by $N(Z_{f,\hat{p}}, \sigma_N^2)$, where $\sigma_N = \mathrm{MMD}(\{\theta_i, w_i^{\mathrm{BQ}}\}_{i=1}^N)$, and w_i^{BQ} is the Bayesian Quadrature weight. Since we approximate this weight empirically and denote the corresponding variance by $\hat{\sigma}_N = \mathrm{MMD}(\{\theta_i, \hat{w}_i^{\mathrm{BQ}}\}_{i=1}^N)$. Since Bayesian Quadrature weight is the optimal weight and this means $\sigma_N \leq \hat{\sigma}_N$. Thus we can upper bound Eq.(31) in Briol et al. (2015) by this variance whose weight is approximated by particles. Then, we get the expression of Theorem 8.

5.6.7 Discussion about the choice of the kernel

The choice of the kernel is crucial numerically and theoretically. In convergence proofs, we assumed that within the affine hull $\mathcal{M}$, there exists a ball with center $\hat{\theta}$ and radius R that is included in $\mathcal{M}$. Bach et al. (2012); Briol et al. (2015) proved that for infinite-dimensional RKHS, e.g., RBF kernel, this assumption never holds. Thus, we can only have the sub-linear convergence for RBF kernels in general. However, as pointed in Briol et al. (2015), even if we use RBF kernels, thanks to the rounding in a computer, what we treat in a simulation are finite-dimensional. This holds to our situation, and in the experiments, we observed the linear convergence of our algorithm.

5.6.8 Detailed experimental settings of SPs

We describe the additional explanation about SPs. In the optimization of SPs, to perform on step optimization in a n dimension parameter space, the Nelder-Mead method needs to evaluate the objective function at least $n + 1$ times, and the grid search method needs to evaluate d^n times, where d is the number of grids in one dimension.

The details of the greedy algorithm are elaborated as follows. The distance we want to minimize between n sampled points and the posterior distribution is defined as

$$\mathrm{D} = \sqrt{\frac{1}{n^2} \sum_{i,j=1}^{n} k_0(\theta_i, \theta_j)}. \tag{5.90}$$

Starting from the first MAP data point, the greedy algorithm tries to find a data points that minimize the distance. The greedy algorithm solves the optimization problem

$$\theta_n = \mathrm{argmin}_\theta \frac{k_0(\theta, \theta)}{2} + \sum_{i=1}^{n-1} k_0(\theta_i, \theta), \tag{5.91}$$

for the nth data point, where $k_0(\theta, \theta')$ is the Stein repoducing kernel defined as

$$\begin{aligned} k_0(\theta, \theta') = \nabla_\theta \cdot \nabla_{\theta'} k(\theta, \theta') + \nabla_\theta k(\theta, \theta') \cdot \nabla_{\theta'} \log p(\theta') + \nabla_{\theta'} k(\theta, \theta') \cdot \nabla_\theta \log p(\theta) + \\ k(\theta, \theta') \nabla_\theta \log p(\theta) \nabla_{\theta'} \log p(\theta'). \end{aligned} \tag{5.92}$$

The Gaussian kernel is used for the base kernel $k(\theta, \theta')$.

The details of the Monte Carlo methods are elaborated as follows. The first sample is drawn by performing MAP approximation, for which we looped 100 times. From the second sample, we take the strategy below. First, we uniformly select 20 base points within existing points. Then, we sample 20 points from a Gaussian distribution, whose location is the base point and scale is set to be 1. We resampled the points until the elements of the 20 points all fall in the range $[-1, 1]$. Finally, we evaluate the 20 points and select the one that performs the best. However, the experiment is hardly feasible. Sampling only 4 data points took 3 minutes and the accuracy is only 56%.

5.6.9 Additional toy data experiments

In this section, we give the situation where our method fails. One failure situation is that there are many modes which have same heights. Since our method relies on finding the near MAP point in the first step of the algorithm and approximating the expectation, when there are many modes with same heights, finding the modes will be meaningless to approximate the expectation and then our method will fail.

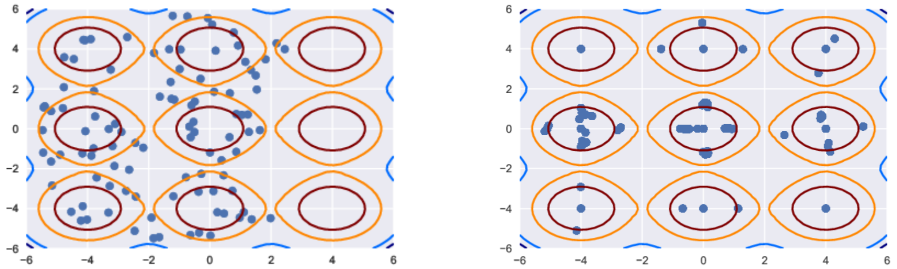
In Figure. 5.8, we consider the two dimensional mixture Gaussian distribution with same heights. There are 9 Gaussian distributions with same heights. We approximate it by our method and SVGD. As shown in Figure. 5.8, neither methods represent the target distribution. Since SVGD is the deterministic method, many particles are trapped in the same mode. In this kind of situation, we need the stochasticity to escape from those local modes.

5.6.10 Additional benchmark dataset experiments

Here, we present that of the protein data in Figure. 5.9.

5.6.11 Discussion about Cache-MMD-FW

As we discussed in Section 5.4.1, we can combine MMD-FW and SVGD. The algorithm is simple. We just replace the Approx-LMO in MMD-FW by Cached approx-LMO as described in Alg. 6. To use the Cached approx-LMO, we first optimize N particles by SVGD. After finishing the SVGD, we store the optimized particles in the “Cache”. Then, in the Cached approx-LMO, in each iteration,



(a) 2D gaussians with same heights approximated by SVGD

(b) 2D gaussians with same heights approximated by MMD-FW

FIGURE 5.8: Toy data example of the mixture Gaussian with same heights and results of MMD-FW and SVGD

we first choose the particle which minimizes the absolute value of $\nabla_\theta \mathrm{MMD}(\theta)^2$ from the Cache. Then we adopt the chosen particle as the initial state of the solution and update it. By doing this, the number of iteration will be drastically small for each iteration. And we eliminate the chosen particle from the cache to prevent from choosing the same particle many times. Based on this Cached approx-LMO, the whole algorithm is given in Alg. 7. We name this algorithm, Cache-MMD-FW. When we use all the particles which are obtained by SVGD, then we will use the usual Approx-LMO in the Algorithm. The theoretical property of this algorithm is as same as the MMD-FW.

Algorithm 6: Cached approx-LMO

```
Input: $\mu_{\hat{p}}^{(k)}$
Output: $k(\cdot, \theta^{L+1})$
$\theta^{(0)} = \mathrm{argmin}_{\theta \in \mathrm{cache}} |\nabla_\theta \mathrm{MMD}(\theta)^2|$
Eliminate the chosen $\theta$ from the Cache
for $l = 0 \ldots L$ do
    Compute $\nabla_\theta \mathrm{MMD}^2$ by Eq.(5.15)
    Update $\theta^{(l+1)} \leftarrow \theta^{(l)} + \epsilon^{(l)} \cdot \nabla_\theta \mathrm{MMD}^2$
end for
```

Algorithm 7: Cached MMD minimization by Frank-Wolfe algorithm

```
... as Alg. 3 , except for the input of step 1
and use the Cached approx-LMO at step 3.
Input: A posterior density $p(\theta)$ and particles $\{\theta_n^{(0)}\}_{n=1}^{n}$
obtained by SVGD
$\bar{g}_n$ =Cached approx-LMO($\mu_{\hat{p}}^{(n)}$)
```

We did the numerical experiment about this algorithm on the toy data which we had explained in the previous section. First, we optimized 200 particles by SVGD. We set the number of iteration $L = 10$ in Cached approx-LMO. The results are shown in Fig 5.10.

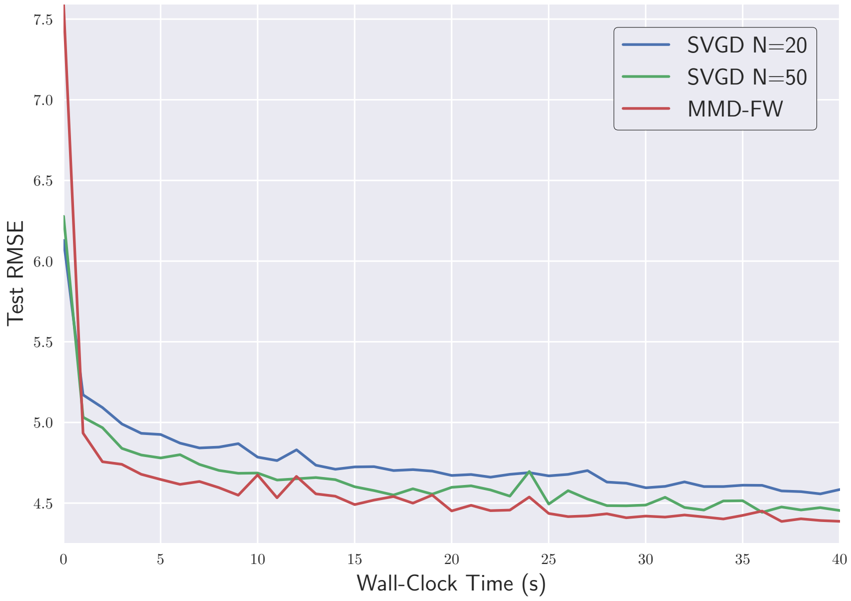


FIGURE 5.9: Comparison of MMD-FW and SVGD in terms of wall clock time with test accuracy(Protein data)

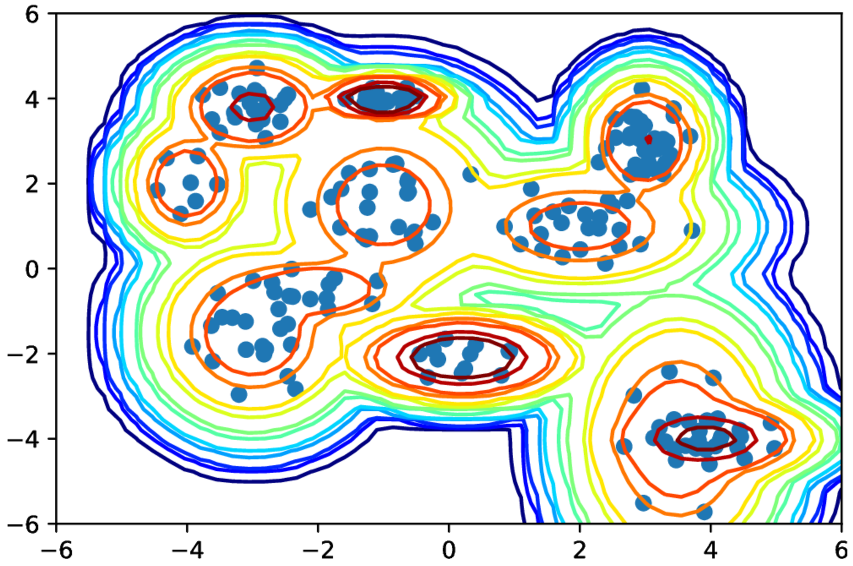


FIGURE 5.10: Toy data results by Cached-MMD-FW

5.6.12 Other variant of FW: Lazy FW algorithm

As we discussed in Section 5.4.1, we can utilize the many variants of the FW to our setting. Here we pick up the Lazy FW Braun et al. (2017).

In Lazy FW, instead of calling the LMO at each step, we re-use the particles which had already been processed and satisfied the criterion. We call such a procedure as Lazy-LMO and shown in Alg 8. This method never improves the sample complexity of the bound, however, it drastically reduces the wall-clock time. When no stored particles satisfy the criterion, we will solve the LMO or update the criterion. When we solve the LMO, we use the Cached approx-LMO of Alg. 6 which contributes to the reduction of the wall clock time of approximate LMO calculation.

Algorithm 8: Lazy LMO

```
Input: $\Phi_n$, K, $\mu_{\hat{p}}^{(n)}$
Output: false or $k(\cdot,\theta)$
if $\theta$ cached with Dg($\mu_{\hat{p}}^{(n)},\theta) \leq -\Phi_n/K$ exists then
   return $k(\theta,\cdot)$\{Cache call\}
else
   $k(\cdot,\theta)$ =Cached approx-LMO($\mu_{\hat{p}}^{(n)}$) of of Alg. 6
   if Dg($\mu_{\hat{p}}^{(n)},\theta) \leq -\Phi/K$ then
      return $k(\theta,\cdot)$ and add $\theta$ to cache
   else
      return false
   end if
end if
```

Algorithm 9: Lazy MMD-FW

```
Input: Accuracy parameter $K$, a posterior density $p(\theta)$,
initial particles $\{\theta_n^{(0)}\}_{n=N}^{n}$ obtained by SVGD
Add all the initial particles into the cache.
$\theta_0 = \mathrm{argmin}_{\theta\in\mathrm{cache}}|\nabla_\theta \ln p(\theta)|$
$\mu_{\hat{p}}^{(0)} = k(\cdot,\theta_0)$
$\Phi_0 = -\min_{\theta\in\mathrm{cache}}$ Dg$(\mu_{\hat{p}}^{(0)},\theta)/2$
for iteration $n$ do
   $\bar{g}_n$ =Lazy-LMO($\Phi_n, K, \mu_{\hat{p}}^{(n)}$)
   if $\bar{g}_n$ =false then
      $\mu_{\hat{p}}^{(n+1)} = \mu_{\hat{p}}^{(n)}$
      $\Phi_{n+1} = \frac{\Phi_n}{2}$
   else
      $\lambda_n = \mathrm{argmin}_{\lambda\in[0,1]} J((1-\lambda)\mu_{\hat{p}}^{(n)} + \lambda\bar{g}_n)$
      Update $\mu_{\hat{p}}^{(n+1)} = (1-\lambda_n)\mu_{\hat{p}}^{(n)} + \lambda_n\bar{g}_n$
      $\Phi_{n+1} = \Phi_n$
   end if
end for
```

To skip the calling of the LMO, we have to calculate the criterion, which is often called the duality gap:

$$\begin{aligned}\mathrm{Dg}(\mu_{\hat{p}}^{(n)},\theta) &:= \langle \mu_{\hat{p}}^{(n)} - \mu_p, \mu_{\hat{p}}^{(n)} - (\theta,\cdot)\rangle \\ &= \sum_{l',l=0}^{n-1} w_{l'}^{(n-1)} w_l^{(n-1)} k(\theta_{l'},\theta_l) - \sum_{l=0}^{n-1} w_l^{(n-1)} \left(k(\theta_l,\theta) + \mu_p(\theta_l)\right) + \mu_p(\theta). \quad (5.93)\end{aligned}$$

The whole algorithm is given in Alg. 9, where we consider the situation that we have already pre-processed particles via SVGD to further reduce the wall clock time. We can also consider the case that particles are not processed by SVGD. In that case, we simply initialize particle sampling from prior or randomly.

Practically, we have to calculate Eq. (5.93) and this is difficult since this includes the integral μ_p. We tried to approximate this term by the technique of biased importance sampling (Bamler et al., 2017), but not work well. Thus, the practical implementation of this algorithm is future work.

The theoretical behavior of Alg. 9 is almost similar to that of ordinary MMD-FW.

I

5.6.13 Discussion about herding and quadrature

When exact integration cannot be done, we often resort to use the quadrature rule approximations. A quadrature rules approximate the integral by weighted sum of functions at the certain points,

$$\hat{Z}_{f,p} = \sum_{n=1}^{N} w_n f(\theta_n), \tag{5.94}$$

where we approximated $p(\theta)$ by $\hat{p}(\theta) = \sum_{n=1}^{N} w_n \delta(\theta_n)$ and $\delta(\theta_n)$ is a Dirac measure at θ_n. There are many ways to specifying the combination of $\{(w_n, \theta_n)\}_{n=1}^{N}$. We call w_ns as *weights* and θ_ns as *particles*. Most widely used quadrature rule is the Monte Carlo(MC). We simply set all the $w_n = \frac{1}{N}$ and we produce θ_ns by drawn from $p(\theta)$ randomly. This non-deterministic sampling based approximation converges at a rate $\mathcal{O}(\frac{1}{\sqrt{N}})$. On the other hand, in the Quasi Monte Carlo, we decide θ_ns to directly minimize the some criterion.

In the kernel herding method (Chen et al., 2010; Bach et al., 2012), the discrepancy measure is the Maximum Mean Discrepancy (MMD). Let $\mathcal{H}$ be a Hilbert space of functions equipped with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and associated norm $\| \cdot \|_{\mathcal{H}}$. The MMD is defined by

$$\mathrm{MMD}(\{(w_n, \theta_n)\}_{n=1}^{N}) = \sup_{f \in \mathcal{H}: \|f\|_{\mathcal{H}} = 1} |Z_{f,p} - \hat{Z}_{f,p}|. \tag{5.95}$$

If we consider $\mathcal{H}$ be a reproducing kernel Hilbert space(RKHS) with a kernel k. In this setup, we can rewrite the MMD using $k(\theta, \theta')$ and set all the $w_i = \frac{1}{N}$,

$$\begin{aligned}
&\mathrm{MMD}^2(\{(w_i = \frac{1}{N}, \theta_i)\}_{i=1}^{N}) \\
&= \sup_{f \in \mathcal{H}: \|f\|_{\mathcal{H}} = 1} |Z_{f,p} - \hat{Z}_{f,p}|^2 = \|\mu_p - \mu_{\hat{p}}\|_{\mathcal{H}}^2 \\
&= \mathrm{Const.} - 2 \iint k(\theta, \theta') p(\theta) \hat{p}(\theta') d\theta d\theta' + \\
&\qquad \iint k(\theta, \theta') \hat{p}(\theta) \hat{p}(\theta') d\theta d\theta' \\
&= \mathrm{Const.} - \frac{2}{N} \sum_{n=1}^{N} \int k(\theta, \theta_n) p(\theta) d\theta + \frac{1}{N^2} \sum_{n,m=1}^{N} k(\theta_n, \theta_m),
\end{aligned} \tag{5.96}$$

where $\mu_p = \int k(\cdot, \theta)p(\theta)d\theta \in \mathcal{H}$. The herding algorithm greedily minimize the above discrepancy in the following way,

$$\begin{aligned}\theta_{N+1} \leftarrow \arg\min_{\theta}[\mathrm{MMD}^2(\{(w_n = \frac{1}{N+1}, \theta_n)\}_{n=1}^N, (w_{N+1} = \frac{1}{N+1}, \theta))] \\ = \arg\max_{\theta}[\frac{2}{N+1}\int k(\theta, \theta')p(\theta')d\theta' - \frac{2}{N+1}\sum_{n=1}^{N} k(\theta, \theta_n)]. \qquad (5.97)\end{aligned}$$

It is widely known that, under certain assumption, they converges at a rate $\mathcal{O}(\frac{1}{N})$.

Chapter 6

Conclusion

In this chapter, we present conclusions and discuss future research directions of this dissertation.

6.1 Conclusions

6.1.1 Expectation Propagation for t-Exponential Family Using q-Algebra

In Chapter 3, we enabled the *t-exponential family* to inherit the important property of the exponential family whose calculation can be efficiently performed thorough natural parameters by using the *q-algebra*. With this natural parameter based calculation, we developed EP for the t-exponential family by introducing the *t-factorization* approach. The key concept of our proposed approach is that the t-exponential family has *pseudo additivity*. When $t = 1$, our proposed EP for the t-exponential family is reduced to the original EP for the ordinary exponential family and t-factorization yields the ordinary data-dependent factorization. Therefore, our proposed EP method can be viewed as a generalization of the original EP. Through illustrative experiments, we confirmed that our proposed EP applied to the Bayes point machine can overcome the drawback of ADF, i.e., the proposed EP method is independent of data permutations. We also experimentally illustrated that proposed EP applied to Student-t process classification exhibited high robustness to outliers compared to Gaussian process classification. Experiments on benchmark data also demonstrated superiority of Student-t process.

6.1.2 Variational inference based on robust divergences

In Chapter 4, we proposed outlier robust variational inference based on robust divergences. We can make our estimation robust against outliers without changing models. We also theoretically compared our proposed method with ordinary variational inference by using the influence function. By using the influence function, we can evaluate how much outliers affect our predictions. The analysis showed that the influence of outliers is bounded in our model, but is unbounded by ordinary variational inference in many cases. Further, experiments demonstrated that our method is robust for both input and output related outliers in both regression and classification settings. In addition, our method outperforms ordinary VI on benchmark datasets.

6.1.3 Bayesian posterior approximation via greedy particle optimization

In Chapter 5, we proposed MMD-FW, a novel approximation method for posterior distributions. Our method enjoys empirically good performance and theoretical guarantee simultaneously. In practice, our algorithm is faster than existing methods in terms of wall clock time and works well even in high-dimensional problems. We also provide the theoretical analysis about the convergence rate and the effect of the inexact step sizes of our proposed method.

6.2 Future work

Here, we present the future research directions.

6.2.1 Future direction of the reformulation of Bayesian inference

We discuss the future research about the reformulation of Bayesian inference. The reformulation of Bayesian inference as the optimization problem is gathering attention recently. In the optimization formulation, the objective function is decomposed to the expected loss that is defined by the cross entropy and the regularization term from the prior distribution. In this dissertation, we used robust divergence for the loss function to enhance robustness.

Theoretically, the use of the cross entropy is only validated when the space of feasible models includes the true data generating mechanism and this is called *M-closed* world setting (Bissiri et al., 2016). On the other hand, if the space of feasible models does not include the true model, which is called *M-open* world setting, there is no theoretical validation for using the cross entropy (Bissiri et al., 2016). When the observed data include outliers, then it is less likely that our model can express the mechanism of outliers, and thus, it corresponds to the *M-open* world settings. The method that replaces the cross entropy is called generalized Bayesian inference and extensive studies have been done recently (Bissiri et al., 2016; Jewson et al., 2018; Knoblauch et al., 2018).

Other than robust inference, replacing the cross entropy term is widely used especially in the deep generative model research. The cross entropies are replaced with other loss functions that is suitable to capture the structured information of the observed data. Tolstikhin et al. (2018) replaced the cross entropy with the Wasserstein distance instead of the cross entropy.

To incorporate the structured information, there is an another approach, called *loss-calibrated approximate Bayesian inference* (Lacoste-Julien et al., 2011). This is motivated from Bayesian decision theory, that is, we would like to make an optimal decision given the utility function $u(\theta, h)$ where h denotes a decision. It is known that optimal decision maximize the utility,

$$G_u(h) = \int p(\theta|D)u(\theta, h)d\theta, \tag{6.1}$$

where we take the expectation with respect to the posterior distribution. When we set a loss function $l(\theta, h)$ as $l(\theta, h) = -u(\theta, h)$, then optimal decision minimize the expected loss.

The approach of loss-calibrated variational inference is to approximate the utility as

$$\begin{aligned}\log G_u(h) = \log \int p(\theta|D)u(\theta,h) &= \log \int q(\theta;\lambda)\frac{p(\theta|D)u(\theta,h)}{q(\theta;\lambda)}d\theta \\ &\geq \int q(\theta;\lambda)\log u(\theta,h)d\theta + \int q(\theta;\lambda)\log\frac{p(\theta|D)}{q(\theta;\lambda)}d\theta \\ &= \mathbb{E}_q[\log u(\theta,h)] + L(\lambda) - \log p(D). \end{aligned} \tag{6.2}$$

Thus, the new optimization problem is:

$$\arg\max_{\lambda,h} L(\lambda) + \mathbb{E}_q[\log u(\theta,h)], \tag{6.3}$$

and we iteratively optimize the above objective function with respect to λ and h.

The difference from usual variational inference is that there is an additional regularization term $\mathbb{E}_q[\log u(\theta,h)]$, which captures the information of the utility function. This means that we restrict the space of the feasible approximate posterior distributions by using the information of the utility function. Lacoste-Julien et al. (2011) clarified that this loss-calibration is especially useful for the structured utility functions, and Cobb et al. (2018) applied it to Bayesian neural networks.

Compared to generalized Bayesian inference, this approach tries to incorporate the structured information via regularization term. To incorporate the additional information, regularization approaches are widely used in the field of the maximum entropy (Dudík et al., 2007; Ganchev et al., 2010). It is still unclear to what problems we should replace the cross entropy to other loss functions or add the regularization term about the utility functions or combine both approaches together. Since no comparison or theoretical analysis have been done about it, we need to clarify about this. Another research direction is to explore robust inference for Bayesian decision making based on loss-calibrated approximate inference. Theoretical and numerical comparison between generalized Bayesian inference and the loss-calibrated approach in terms of the structured data and robustness is needed.

6.2.2 Future direction of the particle approximation

Finally, we describe the extension of the particle approximation method. The approximation method proposed in Chapter 5 is not favorable for problems which need a compuIn such a situation, unlike greedy approximation, a simultaneous optimization for all the particles like SVGD is preferable. The drawbacks of SVGD is that it suffers from the lack of theoretical guarantee and when the posterior density is non-convex, many particles of SVGD are collapsed to the same mode empirically, which results in the failure to approximate the posterior.

To solve these problems, a simple extension of SVGD is to treat it as a stochastic process by combining the Langevin dynamics. Here, we express the target distribution as $\pi \propto e^{-U}$. The dynamics for the i-th particle is ($\forall i = 1, \ldots, N$):

$$d\theta_t^i = -\beta^{-1}\nabla_\theta U(\theta_t^i) + \frac{1}{N}\sum_{n=1}^{N}\left[-\nabla_\theta U(\theta_t^n)k(\theta_t^n, \theta_t^i) + \nabla_\theta k(\theta_t^n, \theta_t^i)\right] + \sqrt{2\beta^{-1}}dw_t, \tag{6.4}$$

where θ^i is the i-th particle and β' is some constant. Compared to usual SGLD for the i-th particle ($\forall i = 1, \ldots, N$),

$$d\theta_t^i = -\beta^{-1}\nabla U_t(\theta_t^i)dt + \sqrt{2\beta^{-1}}dw_t, \tag{6.5}$$

there is an additional term:

$$\frac{1}{N}\sum_{n=1}^{N} R(\theta_t^n, \theta_t^i) = \frac{1}{N}\sum_{n=1}^{N}\left[-\nabla_\theta U(\theta_t^n)k(\theta_t^n, \theta_t^i) + \nabla_\theta k(\theta_t^n, \theta_t^i)\right], \tag{6.6}$$

which expresses the repulsion between the i-th particle and others. When the Gaussian kernel is used, according to Ma et al. (2015), the stationary distribution of the joint distribution of all the particles $\{\theta_t^i\}_{i=1}^N \in \mathbb{R}^{dN}$ of Eq.(6.4), Eq.(6.5) are the same.

The natural question is that what is the advantage of introducing the repulsion term R. We present the intuitive discussion about the convergence speed of the joint and marginal distributions of the particle system.

First, we consider the convergence speed of the joint distribution of $\{\theta_t^i\}_{i=1}^N \in \mathbb{R}^{dN}$ to the stationary distribution. For Eq.(6.4), we need to consider the convergence of the whole N-particle system. For Eq.(6.5), the convergence of the joint distribution corresponds to the convergence of N-parallel SGLD. According to the spectral analysis (Duncan et al., 2016; Kaiser et al., 2017), the convergence speed of Eq.(6.4) is faster than Eq.(6.5) under the moderate assumptions about the potential U and the repulsion term R. Thus, introducing the repulsion term improves the convergence speed for the joint distribution.

Next, we consider the convergence speed of the marginal distribution of each particles, that is, we consider the convergence speed for each particle, $\forall i = 1, \ldots, N$, $\{\theta_t^i\} \in \mathbb{R}^d$. The marginal distributions of each particle are the same $\forall i = 1, \ldots, N$ since all the particles are exchangeable with each other. In Eq.(6.5), since each particle is independent with each other, the convergence speed to the stationary can be considered independently. Since the convergence speed strongly depends on the dimension of the particle, and it depends on d which is the dimension of one particle. On the other hand, in Eq.(6.4), we need to consider the coupling of all the particles and then we consider the marginalization. Thus, the related dimension of the process is dN, which is the whole particle system. Thus, in terms of the dimension, introducing the repulsion term makes the convergence speed of the marginal distributions slow.

The above two intuitive discussions raise the question that introducing the repulsion term really accelerate the convergence speed since its behavior is completely different between the joint and marginal distributions. To tackle this question, one promising approach is to use the Mckean-Vlasov process (Eberle et al., 2018; Bolley et al., 2013; Veretennikov, 2006), which is known as the nonlinear stochastic process. In Mckean-Vlasov process, we do not consider the dN-dimensional stochastic process, but we regard the whole stochastic process as N-particle approximation of the original stochastic process:

$$d\theta_t = -\beta^{-1}\nabla_\theta U(\theta_t^i) + \mathbb{E}_{\theta_t'}\left[-\nabla_\theta U(\theta_t')k(\theta_t', \theta_t) + \nabla_\theta k(\theta_t', \theta_t)\right] + \sqrt{2\beta'^{-1}}dw_t. \tag{6.7}$$

The advantage of this direction is that we do not have to treat the dN-dimensional system and only need to treat the particle system as the d-dimensional system. Instead, the error due to the empirical approximation by N particles appears. Thus, we need to control this empirical approximation error. For this control, using Girsanov theorem (Bakry et al., 2013) seems promising tool and we leave this control to the future work. If we successfully control this error, we can say that Eq.(6.4) is superior to parallel SGLD due to the repulsion term since the obtained algorithm optimize all the particles simultaneously with theoretical guarantee.

Another approach to tackle the above question is to use the functional inequalities (Bakry et al., 2013). For example, the Poincare inequality and the logarithmic Sobolev inequality are used to derive the exponential convergence of the Markov process to the stationary measure (Raginsky et al., 2017). Here, we express the measure which is induced by the stochastic process at time t as μ_t. We say that π satisfies the Poincare inequality with a constant c if π if it satisfies for all $\mu \ll \pi$

$$\chi^2(\mu\|\pi) \leq c\,\mathcal{E}\left(\sqrt{\frac{d\mu}{d\pi}}\right), \tag{6.8}$$

where $\mathcal{E}(g)$ is the Dirichlet form which is defined as

$$\mathcal{E}(g) := \int_{\mathbb{R}^d} \|\nabla g\|^2 d\pi, \tag{6.9}$$

and $\chi^2(\mu\|\pi)$ is the chi-divergence which is defined as

$$\chi(\mu\|\pi) := \int_{\mathbb{R}^d} \left|\frac{d\mu}{d\pi} - 1\right|^2 d\pi. \tag{6.10}$$

Also we say that π satisfies the logarithmic Sobolev inequality with a constant c for all $\mu \ll \pi$,

$$\mathrm{KL}(\mu\|\pi) \leq 2c\,\mathcal{E}\left(\sqrt{\frac{d\mu}{d\pi}}\right). \tag{6.11}$$

It is known that when π satisfies the logarithmic Sobolev inequality, then π also satisfies the Poincare inequality. The consequence of these functional inequalities is the exponential convergence in the variance and the entropy. For example, when π satisfies the logarithmic Sobolev inequality with a constant c, it implies the exponential convergence of the KL divergence with a rate $2/c$ (Theorem 5.2.1 in Bakry et al. (2013)), that is,

$$\mathrm{KL}(\mu_t\|\pi) \leq e^{-2t/c}\mathrm{KL}(\mu_0\|\pi). \tag{6.12}$$

A similar relation holds for the Poincare inequality (Bakry et al., 2013)). Thus, deriving the constant c for those functional inequalities means the deriving the convergence rate. The constant of the Poincare inequality and the logarithmic Sobolev inequality are closely related with each other (Raginsky et al., 2017). Also, the constant of the Poincare inequality is closely related to the spectrum of the generator. For example, when the following stochastic differential equation (SDE) is given

$$d\theta_t = -\beta^{-1}\nabla U(\theta_t)dt + \sqrt{2\beta^{-1}}dw_t, \tag{6.13}$$

we denote the corresponding Markov semigroup as $P = \{P_t\}_{t>0}$ and the Kolmogorov operator as P_s which is defined as

$$P_s f(\theta_t) = \mathbb{E}[f(\theta_{t+s})|\theta_t], \tag{6.14}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a bounded test function. The generator of the associated semigroup is given as

$$\mathcal{L}f(\theta_t) := \lim_{h\to 0^+} \frac{\mathbb{E}[f(\theta_{t+h})|\theta_t] - f(\theta_t)}{h} = \left(-\beta^{-1}\nabla U(\theta_t)\cdot\nabla + \beta^{-1}\nabla^2\right) f(\theta_t). \tag{6.15}$$

If a constant c satisfies Eq.(6.8), then $1/c \geq \lambda$,

$$\lambda := \inf\left\{\frac{\mathcal{E}(g)}{\int_{\mathbb{R}^d} g^2 d\pi} : g \neq 0, \int_{\mathbb{R}^d} g d\pi = 0\right\}. \tag{6.16}$$

Thus, by studying the spectrum of the Markov diffusion operator, we can get the insight about the convergence speed.

Back to our setting, that is, there is an additional term in the SDE,

$$\gamma(\theta_t) := \frac{1}{N}\sum_{n=1}^{N} R(\theta_t^n, \theta_t^i). \tag{6.17}$$

Then, the corresponding generator is written as

$$\mathcal{L}f(\theta_t) = \left(-\beta^{-1}\nabla U(\theta_t)\cdot\nabla + \gamma(\theta_t)\cdot\nabla + \beta^{-1}\nabla^2\right) f(\theta_t). \tag{6.18}$$

Thus, by studying the eigenvalue of this generator, we can get the insight for the convergence rate. This is the another future direction.

Bibliography

Bach, F., Lacoste-Julien, S., and Obozinski, G. (2012). On the equivalence between herding and conditional gradient algorithms. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1359–1366.

Bakry, D., Gentil, I., and Ledoux, M. (2013). *Analysis and geometry of Markov diffusion operators*, volume 348. Springer Science & Business Media.

Bamler, R., Zhang, C., Opper, M., and Mandt, S. (2017). Perturbative black box variational inference. In *Advances in Neural Information Processing Systems*, pages 5086–5094.

Basu, A., Harris, I. R., Hjort, N. L., and Jones, M. C. (1998). Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559.

Beck, A. and Teboulle, M. (2004). A conditional gradient method with linear rate of convergence for solving convex linear systems. *Mathematical Methods of Operations Research*, 59(2):235–247.

Bierkens, J., Fearnhead, P., Roberts, G., et al. (2019). The zig-zag process and super-efficient sampling for bayesian analysis of big data. *The Annals of Statistics*, 47(3):1288–1320.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Bissiri, P. G., Holmes, C. C., and Walker, S. G. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103–1130.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Bolley, F., Gentil, I., and Guillin, A. (2013). Uniform convergence to equilibrium for granular media. *Archive for Rational Mechanics and Analysis*, 208(2):429–445.

Bonald, T. and Combes, R. (2017). A minimax optimal algorithm for crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 4355–4363.

Braun, G., Pokutta, S., and Zink, D. (2017). Lazifying conditional gradient algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, pages 566–575.

Briol, F.-X., Oates, C., Girolami, M., and Osborne, M. A. (2015). Frank-wolfe bayesian quadrature: Probabilistic integration with theoretical guarantees. In *Advances in Neural Information Processing Systems*, pages 1162–1170.

Chen, W. Y., Mackey, L., Gorham, J., Briol, F.-X., and Oates, C. J. (2018). Stein points. *arXiv preprint arXiv:1803.10161*.

Chen, Y., Welling, M., and Smola, A. (2010). Super-samples from kernel herding. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, pages 109–116.

Cichocki, A., Cruces, S., and Amari, S.-i. (2011). Generalized alpha-beta divergences and their application to robust nonnegative matrix factorization. *Entropy*, 13(1):134–170.

Cobb, A. D., Roberts, S. J., and Gal, Y. (2018). Loss-calibrated approximate inference in bayesian neural networks. *arXiv preprint arXiv:1805.03901*.

Dai, B., He, N., Dai, H., and Song, L. (2016). Provable bayesian inference via particle mirror descent. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 985–994.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Dheeru, D. and Karra Taniskidou, E. (2017). UCI machine learning repository.

Ding, N., Qi, Y., and Vishwanathan, S. (2011). t-divergence based approximate inference. In *Advances in Neural Information Processing Systems*, pages 1494–1502.

Ding, N. and Vishwanathan, S. (2010). t-logistic regression. In *Advances in Neural Information Processing Systems*, pages 514–522.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics letters B*, 195(2):216–222.

Dudík, M., Phillips, S. J., and Schapire, R. E. (2007). Maximum entropy density estimation with generalized regularization and an application to species distribution modeling. *Journal of Machine Learning Research*, 8(Jun):1217–1260.

Duncan, A. B., Lelievre, T., and Pavliotis, G. (2016). Variance reduction using nonreversible langevin samplers. *Journal of statistical physics*, 163(3):457–491.

Eberle, A., Guillin, A., and Zimmer, R. (2018). Quantitative harris-type theorems for diffusions and mckean–vlasov processes. *Transactions of the American Mathematical Society*, pages 7135–7173.

Efron, B., Tibshirani, R., et al. (1998). The problem of regions. *The Annals of Statistics*, 26(5):1687–1718.

Elsayed, G., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., and Sohl-Dickstein, J. (2018). Adversarial examples that fool both computer vision and time-limited humans. In *Advances in Neural Information Processing Systems*, pages 3910–3920.

Fienberg, S. E. et al. (2006). When did bayesian inference become" bayesian"? *Bayesian analysis*, 1(1):1–40.

Fujisawa, H. and Eguchi, S. (2008). Robust parameter estimation with a small bias against heavy contamination. *Journal of Multivariate Analysis*, 99(9):2053 – 2081.

Fushiki, T. et al. (2005). Bootstrap prediction and bayesian prediction under misspecified models. *Bernoulli*, 11(4):747–758.

Ganchev, K., Gillenwater, J., Taskar, B., et al. (2010). Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(Jul):2001–2049.

Gelfand, A. E. and Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741.

Germain, P., Bach, F., Lacoste, A., and Lacoste-Julien, S. (2016). Pac-bayesian theory meets bayesian inference. In *Advances in Neural Information Processing Systems*, pages 1884–1892.

Ghahramani, Z. and Rasmussen, C. E. (2003). Bayesian monte carlo. In *Advances in neural information processing systems*, pages 505–512.

Ghosh, A. and Basu, A. (2016). Robust bayes estimation using the density power divergence. *Annals of the Institute of Statistical Mathematics*, 68(2):413–437.

Giordano, R., Broderick, T., and Jordan, M. (2015). Robust inference with variational bayes. *arXiv preprint arXiv:1512.02578*.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET.

Grenander, U. and Miller, M. I. (1994). Representations of knowledge in complex systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(4):549–603.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773.

Guélat, J. and Marcotte, P. (1986). Some comments on wolfe's ‘away step’. *Mathematical Programming*, 35(1):110–119.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications.

Hernández-Lobato, D. and Hernández-Lobato, J. M. (2016). Scalable gaussian process classification via expectation propagation. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 168–176.

Hernandez-Lobato, J., Li, Y., Rowland, M., Bui, T., Hernandez-Lobato, D., and Turner, R. (2016). Black-box alpha divergence minimization. In *Proceedings of the 33th International Conference on Machine Learning*, pages 1511–1520.

Hinton, G. and Van Camp, D. (1993). Keeping neural networks simple by minimizing the description length of the weights. In *Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory*.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

Huber, P. and Ronchetti, E. (2011). *Robust Statistics*. Wiley Series in Probability and Statistics. Wiley.

Huszár, F. and Duvenaud, D. (2012). Optimally-weighted herding is bayesian quadrature. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*, pages 377–386.

Jaakkola, T. S. and Jordan, M. I. (2013). Computing upper and lower bounds on likelihoods in intractable networks. *arXiv preprint arXiv:1302.3586*.

Jaggi, M. (2013). Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pages 427–435.

Jeffreys, H. (1939). Theory of probability clarendon press.

Jewson, J., Smith, J., and Holmes, C. (2018). Principles of bayesian inference using general divergence criteria. *Entropy*, 20(6):442.

Johnson, M. J., Duvenaud, D. K., Wiltschko, A., Adams, R. P., and Datta, S. R. (2016). Composing graphical models with neural networks for structured representations and fast inference. In *Advances in neural information processing systems*, pages 2946–2954.

Jylänki, P., Vanhatalo, J., and Vehtari, A. (2011). Robust gaussian process regression with a student-t likelihood. *Journal of Machine Learning Research*, 12(Nov):3227–3257.

Kaiser, M., Jack, R. L., and Zimmer, J. (2017). Acceleration of convergence to equilibrium in markov chains by breaking detailed balance. *Journal of Statistical Physics*, 168(2):259–287.

Kim, H.-C. and Ghahramani, Z. (2008). Outlier robust gaussian process classification. *Structural, Syntactic, and Statistical Pattern Recognition*, pages 896–905.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Knoblauch, J., Jewson, J. E., and Damoulas, T. (2018). Doubly robust bayesian inference for non-stationary streaming data with \beta-divergences. In *Advances in Neural Information Processing Systems*, pages 64–75.

Koh, P. W. and Liang, P. (2017). Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1885–1894.

Konishi, S. and Kitagawa, G. (1996). Generalised information criteria in model selection. *Biometrika*, 83(4):875–890.

Koyejo, O. and Ghosh, J. (2013). Constrained bayesian inference for low rank multitask learning. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, pages 341–350.

Lacoste-Julien, S., Huszár, F., and Ghahramani, Z. (2011). Approximate inference for the loss-calibrated bayesian. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 416–424.

Lacoste-Julien, S. and Jaggi, M. (2015). On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pages 496–504.

Lacoste-Julien, S., Jaggi, M., Schmidt, M., and Pletscher, P. (2013). Block-coordinate frank-wolfe optimization for structural svms. In *Proceedings of the 30th International Conference on Machine Learning*, pages 53–61.

Lacoste-Julien, S., Lindsten, F., and Bach, F. (2015). Sequential kernel herding: Frank-wolfe optimization for particle filtering. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, pages 544–552.

Li, Y. and Gal, Y. (2017). Dropout inference in bayesian neural networks with alpha-divergences. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2052–2061.

Li, Y. and Turner, R. E. (2016). Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, pages 1073–1081.

Lichman, M. (2013). UCI machine learning repository.

Lindley, D. V. and Smith, A. F. (1972). Bayes estimates for the linear model. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(1):1–18.

Liu, Q. (2017). Stein variational gradient descent as gradient flow. In *Advances in neural information processing systems*, pages 3118–3126.

Liu, Q., Lee, J., and Jordan, M. (2016). A kernelized stein discrepancy for goodness-of-fit tests. In *Proceedings of the 33th International Conference on Machine Learning*, pages 276–284.

Liu, Q., Peng, J., and Ihler, A. T. (2012). Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 692–700.

Liu, Q. and Wang, D. (2016). Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2378–2386.

Locatello, F., Khanna, R., Ghosh, J., and Rätsch, G. (2017a). Boosting variational inference: an optimization perspective. *arXiv preprint arXiv:1708.01733*.

Locatello, F., Khanna, R., Tschannen, M., and Jaggi, M. (2017b). A unified optimization view on generalized matching pursuit and frank-wolfe. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 860–868.

Ma, Y.-A., Chen, T., and Fox, E. (2015). A complete recipe for stochastic gradient mcmc. In *Advances in Neural Information Processing Systems*, pages 2917–2925.

McNeish, D. (2016). On using bayesian methods to address small sample problems. *Structural Equation Modeling: A Multidisciplinary Journal*, 23(5):750–773.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

Metropolis, N. and Ulam, S. (1949). The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341.

Minka, T. P. (2001). *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology.

Murphy, K. P. (2012). Machine learning: A probabilistic perspective.

Narayan, K., Punjani, A., and Abbeel, P. (2015). Alpha-beta divergences discover micro and macro structures in data. In *Proceedings of the 32th International Conference on Machine Learning*, pages 796–804.

Neal, R. M. et al. (2011). Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11).

Nivanen, L., Le Mehaute, A., and Wang, Q. A. (2003). Generalized algebra within a nonextensive statistics. *Reports on Mathematical Physics*, 52(3):437–444.

Pearl, J. (1982). Reverend bayes on inference engines: a distributed hierarchical approach. In *Proceedings of the Second AAAI Conference on Artificial Intelligence*, pages 133–136. AAAI Press.

Raginsky, M., Rakhlin, A., and Telgarsky, M. (2017). Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory*, pages 1674–1703.

Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, pages 814–822.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian processes for machine learning*, volume 1. MIT press Cambridge.

Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., and Moy, L. (2010). Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322.

Robert, C. and Casella, G. (2011). A short history of markov chain monte carlo: Subjective recollections from incomplete data. *Statistical Science*, pages 102–115.

Rukhin, A. L. (1990). Kolmogorov's contributions to mathematical statistics. *The Annals of Statistics*, 18(3):1011–1016.

Samek, W., Blythe, D., Müller, K.-R., and Kawanabe, M. (2013). Robust spatial filtering with beta divergence. In *Advances in Neural Information Processing Systems*, pages 1007–1015.

Saul, L. K., Jaakkola, T., and Jordan, M. I. (1996). Mean field theory for sigmoid belief networks. *Journal of artificial intelligence research*, 4:61–76.

Seeger, M. (2005). Expectation propagation for exponential families. Technical report.

Shah, A., Wilson, A., and Ghahramani, Z. (2014). Student-t processes as alternatives to gaussian processes. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, pages 877–885.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244.

Suyari, H. and Tsukada, M. (2005). Law of error in tsallis statistics. *IEEE Transactions on Information Theory*, 51(2):753–757.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schölkopf, B. (2018). Wasserstein auto-encoders. In *International Conference on Learning Representations*. OpenReview. net.

Van der Vaart, A. W. (1998). *Asymptotic statistics*, volume 3. Cambridge university press.

Veretennikov, A. Y. (2006). On ergodic measures for mckean-vlasov stochastic equations. In *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 471–486. Springer.

Wang, H. and Yeung, D.-Y. (2016). Towards bayesian deep learning: A survey. *arXiv preprint arXiv:1604.01662*.

Wang, Y., Kucukelbir, A., and Blei, D. M. (2017). Robust probabilistic modeling with bayesian data reweighting. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3646–3655.

Wang, Z., Ren, T., Zhu, J., and Zhang, B. (2019). Function space particle optimization for bayesian neural networks. In *International Conference on Learning Representations*.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pages 681–688.

Zellner, A. (1988). Optimal information processing and bayes's theorem. *The American Statistician*, 42(4):278–280.

Zhang, Y., Chen, X., Zhou, D., and Jordan, M. I. (2016). Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *Journal of Machine Learning Research*, 17(1):3537–3580.

Zhu, J., Chen, N., and Xing, E. P. (2014). Bayesian inference with posterior regularization and applications to infinite latent svms. *Journal of Machine Learning Research*, 15:1799–1847.